Toposław – a lexicographic framework for multi-word units

Małgorzata Marciniak*, Agata Savary†*, Piotr Sikora*, Marcin Woliński*

*Institute of Computer Science, Polish Academy of Sciences,
J.K. Ordona 21, 01-237 Warszawa, Poland
{malgorzata.marciniak, wolinski}@ipipan.waw.pl
ps237668@students.mimuw.edu.pl

† Université François Rabelais Tours,
3, pl. Jean-Jaurès, 41029 Blois, France
agata.savary@univ-tours.fr

Abstract. The paper presents a tool for the creation of an electronic dictionary of multi-word proper names. *Toposław* uses graphs for the representation of inflectional and pragmatic variants of names. It cooperates with *Morfeusz*, a morphological analyser and generator for Polish words, and *Multiflex*, a cross-language morpho-syntactic generator of multi-word units. Our goal was to create a user-friendly tool that makes a lexicographic work easy and efficient. In the paper we describe facilities for graph creation, management and debugging. The presented tool was applied to create a dictionary of Warsaw urban proper names.

Keywords: electronic dictionary, lexicographic framework, graphs, urban proper names, Polish

1 Introduction

Proper names and other named entities are of crucial quantitative and qualitative importance for natural language processing (NLP) due to their frequent occurrence in texts and their rich semantic content. Despite continual efforts in the NLP community aiming at named entity extraction and modelling, the formal linguistic description of proper names remains a challenge.

We are interested in a lexical description of multi-word names in a particular application domain: the urban transportation system in Warsaw. In [1] we present a project of creating a dictionary of Polish toponyms relevant to Warsaw transportation. The project includes the development of a lexicographic framework, *Toposław*, that allows us the creation of the dictionary in an efficient and quality-ensured manner. *Toposław* cooperates with: (i) *Morfeusz SGJP* (a new version of *Morfeusz* [2]), a morphological analyzer and generator for Polish single words, (ii) *Multiflex* [3], a cross-language, morpho-syntactic generator of multi-word units built upon *Unitex* [4], (iii) an enhanced graph editor based on *Unitex*.

The first version of the tool described in [5] was the starting point of the lexicographer's work. On the basis of these first experiences we identified several obstacles to effective lexicographic work: the growing number of graphs describing proper names, their complexity, and the need for graph reusing facilities. The enhanced framework, including solutions to these problems, is presented in the paper.

Our dictionary contains 8,935 Warsaw names: streets, monuments, buildings, etc. One of its main goals is the recognition and generation of all grammatical forms and variants of a proper name, including transliterated spoken variants. For example, the full official name of *ulica Bitwy Warszawskiej 1920 r.* 'The Battle of Warsaw 1920 Street' is abbreviated in practice into *ul. Bitwy Warszawskiej* 'The Battle of Warsaw Str.', while for transliterated speech processing the year should be spelled out: *tysiąc dziewięćset dwudziestego roku* 'year nineteen twenty'. The number of such inflected forms and variants for all entries in our dictionary exceeds 309,000.

As mentioned, our dictionary is meant for knowledge-based analysis and generation of written and spoken Polish texts. According to [6], knowledge-poor methods give good results in matching two-component person names in Polish texts (up to 0.99 accuracy in some cases). An automatic lemmatization of such names is more problematic (below 0.67 accuracy in case of component inversion). We expect such methods to be even less effective for names, allowing for omissions, inversions and other transformations. We believe that in order to achieve high-quality analysis and generation of urban names we need to use knowledge-based methods.

A Polish named-entity extraction tool, based on rich gazetteers (164,000 names), local grammars (198 rules) and string-based similarity is described in [7]. It allows for the population of an ontology from free-form text. Our approach is complementary in the sense that we describe not only morphological but also syntactic and semantic variants, and we point out pragmatic variants necessary for text generation. Our grammars (i.e. graphs) are fully lexicalized, which avoids both noise and silence.

As far as lexicographic frameworks covering the morpho-syntax of multi-word units in other languages are concerned, the most notable example we are aware of is *WS4LR* [8], later renamed *LeXimir*, developed for Serbian, a language of a morphological complexity comparable to Polish. Like *Toposław* it includes *Multiflex* and submodules of *Unitex*. It also contains some facilities for rule-based automatic graph prediction which speed up the lexicographer's work [9]. A partial conflation of these facilities with our advanced graph management methods for Polish is currently being considered.

The organization of the paper is as follows. The next section discusses the representation of objects in our lexicographic tool. Section 3 explains how graphs are used for the representation of inflection of multi-word units and their variants. The following section is devoted to graph management and debugging. Section 5 includes information on the filtering of names according to various criteria. Finally, conclusions and perspective of the tools' development are presented.

2 Objects in *Toposław*

The core entity in any dictionary created with *Toposław* is an object. For the dictionary of Warsaw urban proper names, these are city objects like streets, stops, buildings etc. Each object is linked with its name(s) and with some semantic and pragmatic information.

Objects are characterised with semantic concepts represented in a hierarchy. For the application of *Toposław* to Warsaw toponyms we have defined a rather simple hierarchy of concepts. It consists of: (i) two supertypes – PLACE and PERSONAGE (person names are described as individual entries because each of them can be embedded in several city names), (ii) six subtypes of type PLACE: ADMINISTRATIVE AREA, ROAD, COMMUNICATION POINT, FACILITY, HYDRONYM and MONUMENT (iii) nine subtypes of the first three (ii) types, e.g., for COMMUNICATION POINT: STOP, RAILWAY STATION and AIRPORT. The hierarchy is described in [1].

We assume that one object can have several names. Such names (as opposed to variants of one name) are described separately and only then linked with one object. For example, in the Warsaw urban proper names dictionary, the names *Plac Thomasa Woodrowa Wilsona* 'Thomas Woodrow Wilson Square' and *Plac Komuny Paryskiej* 'Parisian Commune Square' refer to the same object, but the first is its current, official name, while the second is its former name. Each name is characterised by one of four stylistic labels defined in *Toposław*: *base*, *common*, *former*, or *marked*.

Moreover, one name can by attached to several objects. For example most Polish towns have a street named *Szkolna* 'School street'. So if we want to represent the Warsaw agglomeration with its satellite towns, we attach the same name to several objects representing streets in different towns. It is possible to add a comment to the object to specify for example the administrative area to which different objects bearing the same name belong.

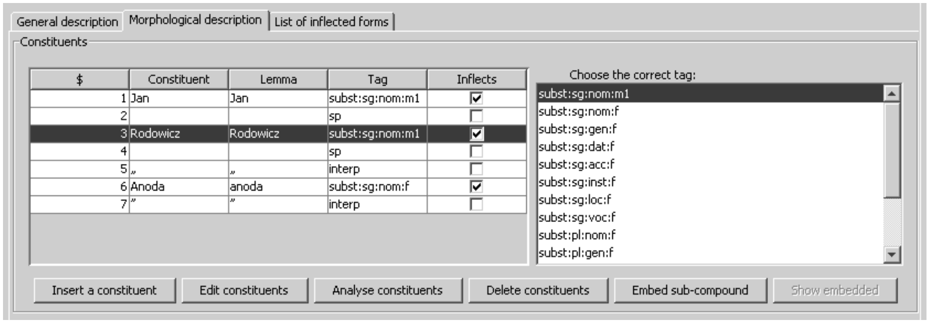
3 Morphology and Variants of a Name

Polish is a language of a rather rich inflectional morphology, single and compound nouns inflect for case and thus usually have at least 7 forms. As shown in [10], the case inflection of names can be combined with different types of variants based on abbreviations, acronyms, initials, numerals' spelling, component omission or inversion, etc. Example (1) illustrates variants of a person name (in nominative) consisting of a first name, a last name and a pseudonym. The pseudonym can appear before or after the last name, and can be in quotes or not, the forename can be abbreviated or missing, etc. All these variants can appear in any of the 7 cases. As a result, the full list of variants contains dozens of forms, and our dictionary should describe them all.

(1) Jan Rodowicz "Anoda", Jan Rodowicz Anoda, J. Rodowicz "Anoda", J. Rodowicz Anoda, Rodowicz "Anoda", Rodowicz Anoda, Jan "Anoda" Rodowicz, Jan Anoda Rodowicz, J. "Anoda" Rodowicz, J. Anoda Rodowicz, "Anoda" Rodowicz, Anoda Rodowicz, Jan Rodowicz, J. Rodowicz, Rodowicz

3.1 Morphological Description of Components

Clearly, most variants of a multi-word unit can be obtained by combinations of inflected forms of their components. Therefore the morphological description of individual words, both common words and proper names, is a necessary prerequisite for our dictionary. In *Toposław* this description is ensured by *Morfeusz SGJP*, a morphological



$	Constituent	Lemma	Tag	Inflects
1	Jan	Jan	subst:sg:nom:m1	✓
2			sp	☐
3	Rodowicz	Rodowicz	subst:sg:nom:m1	✓
4			sp	☐
5	„	„	interp	☐
6	Anoda	anoda	subst:sg:nom:f	✓
7	”	”	interp	☐

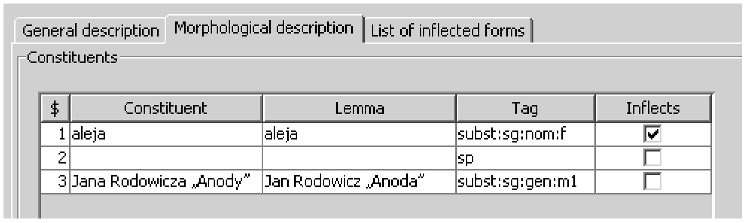
Choose the correct tag:

subst:sg:nom:m1
subst:sg:nom:f
subst:sg:gen:f
subst:sg:dat:f
subst:sg:acc:f
subst:sg:inst:f
subst:sg:loc:f
subst:sg:voc:f
subst:pl:nom:f
subst:pl:gen:f

Fig. 1. The labelled lemma for the name *Jan Rodowicz „Anoda”*

analyser and generator for Polish single words based on the data of the *Grammatical Dictionary of Polish* [11]. For the needs of our dictionary *Morfeusz SGJP* (further called *Morfeusz*) has been extended with 1612 entries: 1005 nouns (not surprisingly last names and geographical names) and 607 adjectives (mainly adjectives derived from geographical names, e.g., *kabacki* from *Kabaty*, but also some other less frequently used adjectives like *arbuzowy*, an adjective from *watermelon*).

The *Morfeusz* analyzer provides all possible morphological interpretations of a word. This allows us to label the constituents of a compound name with their lemmas and morphological features. As shown in Fig. 1 (in the *Constituents* panel), the name is first tokenized and tokens are numbered, whereas separators (blank spaces, hyphens, etc.) and non alphabetical items (quotes, numbers, etc.) are considered as separate tokens. Then, the tokens are analyzed by *Morfeusz* and if any of them have multiple interpretations, the operator has to select the appropriate one. Here, the interpretation selected for *Rodowicz* is substantive, singular, nominative, masculine human (*subst:sg:nom:m1*).



$	Constituent	Lemma	Tag	Inflects
1	aleja	aleja	subst:sg:nom:f	✓
2			sp	☐
3	Jana Rodowicza „Anody”	Jan Rodowicz „Anoda”	subst:sg:gen:m1	☐

Fig. 2. The name *Aleja Jana Rodowicza „Anody”* ‘Jan Rodowicz “Anoda” avenue’ after *Jana Rodowicza „Anody”* has been marked as a sub-compound

The operator can also mark a fragment of the name as a sub-compound to be described separately. We use this mechanism for compound names of persons, which tend to occur in several urban names. For instance in Fig. 2 seven tokens of a street name have been grouped in order to create a single compound component *Jana Rodowicza “Anody”* which corresponds to the person name entry discussed above. Delimiting embedded structures within long names such as *Aleja Jana Rodowicza „Anody”* ‘Jan Rodowicz “Anoda” avenue’ allows for a better modularity and compactness of description.

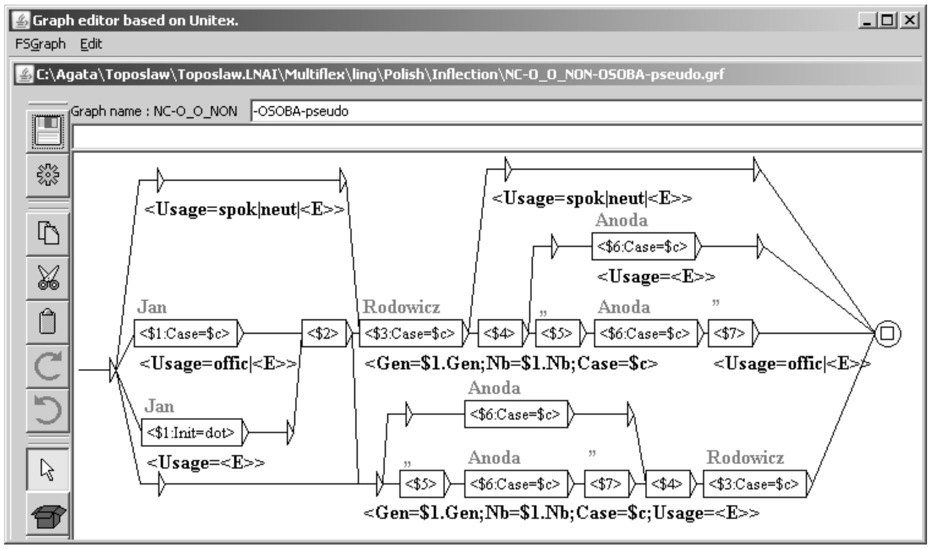


Fig. 3. Inflection graph for the name *Jan Rodowicz „Anoda”*

3.2 Inflection Graphs

Once the components are morphologically identified, the inflection and variation of compound names is modelled in our dictionary using *inflection graphs* defined within *Multiflex* [3]. Fig. 3 shows the inflection graph of the name whose variants are shown in (1). The leftmost triangle represents the entry point of the graph, while the circle enclosing a square shows its exit. The numbered boxes correspond to constituents of the name (words, spaces, punctuation or sub-compounds). The arrow-laden lines that connect the boxes represent various paths which can be used while generating the inflected forms of a name. Here, the bottom-most path describes the elliptic variant with a word order change *„Anoda” Rodowicz.*

The formulae inside boxes consist of constituents’ indexes and equations on morphological variables. These equations impose constraints on the inflection and agreement of constituents. For example, the equation *Case=$c* means that the component inflects for case. When this variable reoccurs the respective components must agree in case, as in the case of components *$1*, *$3* and *$6* in the middle path of Fig. 3. The formulae appearing below paths determine the features of the inflected forms of the whole compound as a function of the features of its constituents. For instance in the bottom-most path on Fig. 3 the resulting forms inherit their gender and number from the first constituent and have the conforming case (*Gen=$1.Gen;Nb=$1.Nb;Case=$c*). The meaning of the final feature *Usage*=⟨*E*⟩ is explained below.

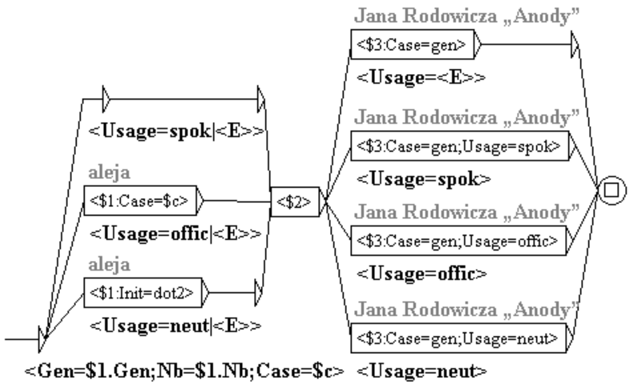


Fig. 4. Inflection graph for the name *aleja Jana Rodowicza „Anody"* 'Jan Rodowicz "Anoda" Avenue' with an embedded multi-word component

Recall that an embedded name can be marked as a sub-compound within a longer name. In this case the sub-compound is referred to as a single component, which increases the modularity of the description since person names are described once for all their occurrences in larger names. Moreover, the number of inflection graphs is kept relatively low (e.g. street, square, and avenue names containing structurally different person names can be covered by one graph) and their complexity decreases. For instance, Fig. 4 shows the graph for the compound whose components are depicted in Fig. 2. If the person name were not delimited as a subcompound, the graph couldn't be applied to other street names containing different numbers of components such as *ulica Kazimierza Pułaskiego* 'Kazimierz Pułaski Street'.

Features All features that can be used to characterise compound inflected forms described by a graph are divided into three types shown in Fig. 5:

- features shared with the underlying module for simple words, here 12 categories and 38 values appearing in the *Morfeusz*' tagset (see ⟨CATEGORIES⟩),
- features freely defined by the user on the level of multi-word units, such as pragmatic labels described below (see ⟨EXTRA_CATEGORIES⟩),
- features implemented in *Multiflex* in order to handle morpho-graphical problems, such as letter case, initialisms and acronyms (see ⟨GRAPHICAL_CATEGORIES⟩).

Pragmatic Variants Among many variants of a name, we want to distinguish a few important pragmatic variants marked by specific values of the *Usage* category:

- the official variant (*offic*) used in official lists and documents,

⟨CATEGORIES⟩

Nb:	sg, pl
Case:	nom, gen, dat, acc, inst, loc, voc
Gen:	m1, m2, m3, f, n1, n2, p1, p2, p3
Pers:	pri, sec, ter
Deg:	pos, com, sup
Asp:	imperf, perf
. . .	

⟨EXTRA_CATEGORIES⟩

Usage:	⟨E⟩, offic, neut, spok

⟨GRAPHICAL_CATEGORIES⟩

LetterCase:	same, all_lower, all_upper, first_upper, first_upper_each_word, no_letter_case, other
Init:	⟨E⟩,dot,no_dot,dot2,no_dot2,dot3,no_dot3, dot4, no_dot4, dot5, no_dot5

Fig. 5. Three kinds of categories in the morphological model of Polish in *Multiflex*

- the neutral variant (*neut*) preferred in text generation,
- the neutral spoken variant (*spok*) preferred for speech generation.

In Fig. 3 the top-most path describes the elliptical variant *Rodowicz* annotated as the neutral spoken (*spok*), neutral (*neut*), or unmarked (⟨*E*⟩) variant. The bottom-most path allows us to obtain the elliptical variant *„Anoda" Rodowicz* described as unmarked. In order to compactly represent several identical forms with different feature values, a feature structure can contain alternative values. Here, the alternative operator '|' allows us to reduce the graph's size from 26 to 15 paths. Note that two different values of the same category cannot be selected on the same path. Here for instance the form *Jan Anoda Rodowicz* can be generated with the unmarked value of *Usage* but not with *offic* because the path omitting quotes (i.e. components *$5* and *$7*) has the constraint ⟨*Usage*=⟨*E*⟩⟩.

Initials and Letter Case Urban proper names frequently take abbreviated forms of their components when appearing in written texts. Any first name can be reduced to its one or two initial letters followed by a dot as in example (1). Similar behavior can be noted in the words 'Street', 'Square', 'Avenue', as well as titles and functions such as 'General'.

Using a dictionary of abbreviations would be most appropriate, unfortunately we are not aware of such a dictionary for Polish. Thus we propose the following partial solution. Whenever an abbreviation is constructed from one to five initial letters, whether or not followed by a dot, the category *Init* (mentioned in Fig. 5) with the corresponding value can be used. For instance in Fig. 3 component *$1* (*Jan*) can be replaced by its initial (*J.*) due to the equation *Init=dot*.

Note that some words are abbreviated differently than by a prefix, as in *płk* for *pułkownik* 'colonel'. Moreover abbreviations or acronyms may be formed from inflected words as *W-wie* for *Warszawie*, or may become independent inflecting lexemes, e.g. *ONZ*, *ONZ-u* for *Organizacja Narodów Zjednoczonych* 'United Nations Organization'. Such cases are currently described with specific dedicated inflectional graphs.

Many urban names contain capitalized common words, as for instance *ulica* ***Długa*** 'Long Str.' or ***Most*** *Syreny* 'Siren Bridge' These words are described in Morfeusz in lower case only. Thus, when such components are morphologically analyzed they obtain lower case lemmas. For instance in Fig. 1 the nickname *Anoda* is assigned a common word lemma *anoda* 'anode'. In order to express this difference in spelling between the lemma and its form we have introduced the *LetterCase* category (see Fig. 6) mentioned in Fig. 5. It indicates how to transform the letter case of the lemma into the form desired in the dictionary. If no transformation is needed the value is *same*.

The *LetterCase* value is most often implicit, i.e. it does not appear in inflection graphs but is automatically deduced from each component of a compound during its morphological analysis. It can however also be explicitly used in graphs if needed.

Form	**Lemma**	**LetterCase**
Władysław	*Władysław*	*same*
empik	*EMPiK*	*all_lower*
STUDIO	*studio*	*all_upper*
Długa	*długi*	*first_upper*
Centrum handlowe	*centrum handlowe*	*first_upper*
Centrum Handlowe	*centrum handlowe*	*first_upper_each_word*
1976	*1976*	*no_letter_case*
MarcPol	*Marcpol*	*other*

Fig. 6. Values of the *LetterCase* category

4 Graph Management

Graphs are created and modified using an enhanced graph editor based on *Unitex* [4]. It allows for the creation, connection, filling out and deletion of boxes. A graph can be assigned to one or many entries at a time whenever they are simultaneously selected from the list of names.

As the number of graphs grows with the number of names described, managing graphs becomes difficult. Currently we have over 8,900 names with 451 corresponding graphs. The majority of names use only a few graphs, which are thus easy to remember. For the rest, however, the user needs some support.

4.1 Filtering Graphs

When a lexicographer introduces a new proper name, *Toposław* displays the list of currently defined graphs which have the same number of components as the name in question. This significantly reduces the number of graphs that have to be considered.

Moreover in the process of describing a compound, the user is asked to state for each component whether it inflects or not (see the *Inflects* field in the *Constituents* panel in

Fig. 1). This pattern of inflecting components is again used to filter the list of graphs. Namely, a graph matches a name only if:

- for each component marked as inflecting there exists a corresponding box in the graph marked as inflecting (with an equality on some grammatical feature),
- for each component marked as non-inflecting none of the corresponding boxes allow for inflection.

We plan to extend this mechanism by measuring the morphological similarity of components of a name being considered to the components of names already described. This measure will allow us to rank the graphs and suggest the most promising graph for a given name (see [12] and [9] for other graph prediction facilities).

One of the novel features introduced in our version of Unitex's graph editor is the labeling of boxes in the graph with constituents of the compound. When the user selects a graph from the list, a preview of this graph is displayed. As shown in Fig. 3 and Fig. 4, all boxes in this preview image are labelled with the constituents of the current name, which allows one to check, whether the components fit into the graph.

4.2 Tracing Paths in a Graph

After a graph is assigned to a name, its inflection is validated by generating and checking all possible forms, as shown in Fig. 7. If an erroneous form is generated, the lexicographer has to deduce, how this form was obtained. For some names, however, a high complexity of graphs is inevitable. To simplify the task of debugging graphs *Toposław* highlights the path in the graph corresponding to the form selected in the list of generated forms.

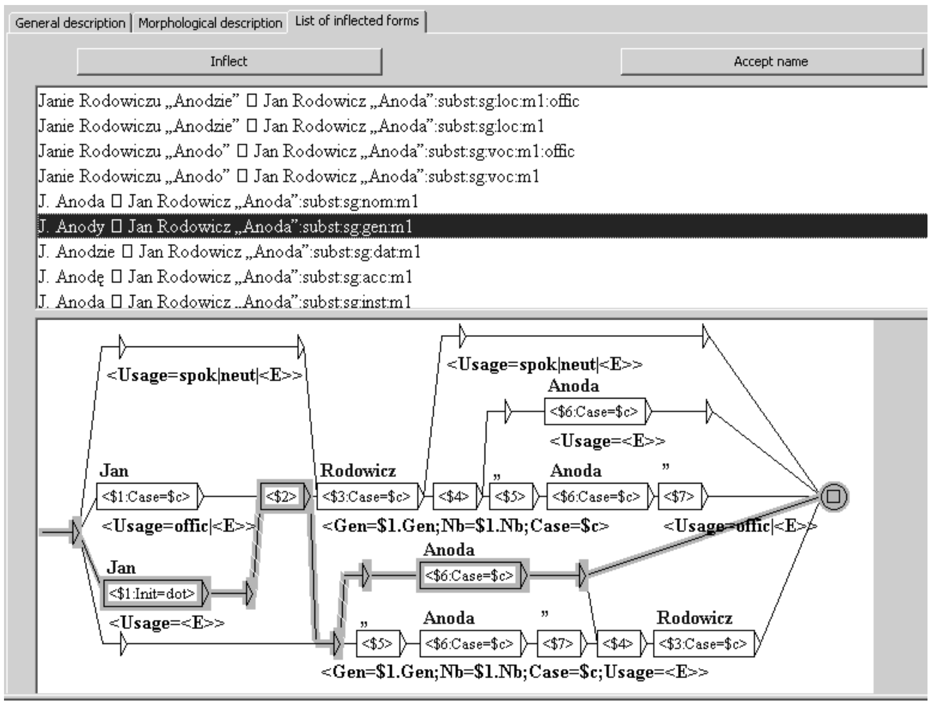
In Fig. 7 the path corresponding to the erroneous form *J. Anody* is highlighted (note the difference with respect to the correct graph in Fig. 3). Automatic modification of the graph based on this information does not seem feasible, but when the path supporting the wrong form is identified, the graph can be corrected rather easily.

If a form can be generated by traversing more than one path, *Toposław* highlights all such paths. In the case of sub-compounds, only the path in the outermost graph is shown, as the simultaneous visualisation of all levels would render the graph unreadable.

The implementation of this feature required changes in *Multiflex*. *Multiflex* uses graphs in a compiled form (minimised and determinised), which has different paths than the graph defined by a lexicographer. To overcome this problem *Toposław*, in visualisation mode, produces a fake compiled form, which is very close to the original graph. This form is used by *Multiflex*, which for each generated form returns the path(s) that generated it. This information is subsequently used by *Toposław* to identify the respective boxes in the original graph.

4.3 New Graphs

A new graph can be created from scratch or based on any existing one. In the former case, graph creation is sped up, in that *Toposław* automatically generates a skeleton



```
Janie Rodowiczu „Anodzie” □ Jan Rodowicz „Anoda”:subst:sg:loc:m1:offic
Janie Rodowiczu „Anodzie” □ Jan Rodowicz „Anoda”:subst:sg:loc:m1
Janie Rodowiczu „Anodo” □ Jan Rodowicz „Anoda”:subst:sg:voc:m1:offic
Janie Rodowiczu „Anodo” □ Jan Rodowicz „Anoda”:subst:sg:voc:m1
J. Anoda □ Jan Rodowicz „Anoda”:subst:sg:nom:m1
J. Anody □ Jan Rodowicz „Anoda”:subst:sg:gen:m1
J. Anodzie □ Jan Rodowicz „Anoda”:subst:sg:dat:m1
J. Anodę □ Jan Rodowicz „Anoda”:subst:sg:acc:m1
J. Anoda □ Jan Rodowicz „Anoda”:subst:sg:inst:m1
```

Fig. 7. The path corresponding to a given inflected form is highlighted in the graph preview

consisting of boxes corresponding to the constituents of the compound for the new graph. Since the lemma of the compound is always supposed to be present among its possible forms, we know for certain that this „backbone” will be needed in the graph.

5 Dictionary Management

Efficient work on a dictionary of several thousand entries requires powerful filtering and exporting capacities.

The list of names visible in the program is dynamic. If a substring is entered in the edit box above the list, only the names containing this substring are displayed. It is also possible to select a graph and display all names the graph is assigned to. Further, the displayed list can be limited to names which: (i) are assigned to a selected concept from the hierarchy, e.g. to streets and squares only, (ii) carry a particular stylistic label (*base*, *common*, *former*, or *marked*), (iii) refer to more than one object.

In order to distinguish already completed dictionary entries from those which are still non-validated, a green or a red icon is displayed next to a name in the former and the latter case, respectively. No icon means that the name has been entered but not yet described at all. An appropriate filter allows to restrict the list of visible entries only to those of a selected status (validated, non-validated, or untouched). More fine-grained filters can limit the list to names only with either unassigned object or graph, or without a morphological description. Finally, one can also view all entries with inconsistent graphs (i.e. whose set of inflected components does not fit the graph's content). All above types of filters can be freely combined.

When the dictionary is being worked on within *Toposław* it is stored as an object-oriented database in a proprietary binary format. When the whole dictionary or a part of it has been completed and validated it can be exported to one of the two interchange formats: a textual *Multiflex*-bound list called a DELAC, or an XML file containing the entries, the morphological analysis of their constituents, and the graphs. Graphs can also be exported as separate files in the *Unitex* format.

6 Conclusions and Perspectives

We have presented *Toposław*, a lexicographic framework meant for creating a Polish dictionary of compound urban proper names. It allows to morphologically analyse the components of a multi-word name (with the help of an external dictionary, e.g. *Morfeusz*), and to compactly describe numerous inflected forms and variants of the name within one graph. An important asset for effective work is the possibility of marking sub-compounds and describing their inflection and variants once for all compounds in which they occur.

Toposław also offers powerful filters for dictionary entries, as well as graph filtering and path tracing. As far as we know, visual debugging tools for inflection graphs are not available within other frameworks. This mechanism could be used in the context of other *Unitex*-based applications (although some decoupling with *Multiflex* would be necessary). Our platform can support other languages whose morphological modules respect the interface constraints described in [10].

Further improvements are needed in handling numbers, years and dates frequent in urban names. Currently a separate graph is needed for each name containing such elements, as we need to represent the correspondence between the written and the spoken form e.g., 3 and *trzeci* 'third'. A general description of the equivalence between such forms would decrease the number and the complexity of graphs.

Toposław has been designed for the needs of a dictionary of urban toponyms however its facilities are clearly relevant to other types of multi-word units. Currently, *Toposław* is being used for describing general purpose compounds, as well as complex economic and financial terms in Polish. In order to allow the completion of this work, we need to add the facility of importing and editing one or more external concept hierarchies and name-to-object relation types.

In [13] *Toposław* undergoes a comparative analysis with another framework used for a massive description of Polish compounds called *Poleng*. It is claimed that the *Multiflex*' graph formalism is more explicit than *Poleng*, and thus relatively universal and interoperable. Moreover, *Multiflex*, unlike *Poleng*, allows to conflate different types of variants (acronyms, inversions, etc.) within one description. *Poleng* on the other hand accounts not only for nominal and adjectival contiguous compounds, but also for verbal multi-word expressions admitting insertions of external elements. Encoding experiments involving both novice and expert lexicographers are also reported. They show that encoding with *Toposław* is up to 3 times less error prone than with *Poleng*. The encoding speed with *Toposław*, including the creation of some inflection graphs from scratch, ranges from 21 entries per hour for a novice lexicographer to 67 for an expert, if no pragmatic labels (i.e. the *Usage* values) are used in graphs (these labels frequently increase the graphs' complexity). The description with *Poleng*, with predefined inflection patterns, is up to 4 times faster due to implicit language-dependent rules and automatic pre-processing facilities. We think that *Toposław* can offer a comparable efficiency (with no loss in universality) if we integrate methods for graph prediction such as those proposed by [9], which is currently being considered.

References

1. Marciniak, M., Rabiega-Wiśniewska, J., Savary, A., Woliński, M., Heliasz, C.: Constructing an Electronic Dictionary of Polish Urban Proper Names. In: Recent Advances in Intelligent Information Systems, Warszawa, Exit (2009) 233–246
2. Woliński, M.: Morfeusz — a Practical Tool for the Morphological Analysis of Polish. In: Proc. of IIS: IIPWM'06, Springer (2006) 503–512
3. Savary, A.: A formalism for the computational morphology of multi-word units. Archives of Control Sciences **15**(3) (2005) 437–449
4. Paumier, S.: Unitex 2.1. User manual. http://www-igm.univ-mlv.fr/∼unitex (2003)
5. Sikora, P., Woliński, M.: Toposław — a Dictionary Creation Tool. In: Recent Advances in Intelligent Information Systems, Warszawa, Exit (2009) 743–749
6. Piskorski, J., Wieloch, K., Sydow, M.: On knowledge-poor methods for person name matching and lemmatization for highly inflectional languages. Information Retrieval **12**(3) (2009) 275–299
7. Abramowicz, W., Filipowska, A., Piskorski, J., Węcel, K., Wieloch, K.: Linguistic Suite for Polish Cadastral System. In: Proceedings of LREC'06, Genoa, Italy. (2006) 2518–2523
8. Krstev, C., Stanković, R., Vitas, D., Obradović, I.: Workstation for Lexical Resources — WS4LR. In: Proc. of LREC'06. (2006) 1692–1697
9. Krstev, C., Stanković, R., Obradović, I., Vitaš, D., Utvic, M.: Automatic Construction of a Morphological Dictionary of Multi-Word Units. LNAI **6233** (2010) 226–237
10. Savary, A., Rabiega-Wiśniewska, J., Woliński, M.: Inflection of Polish Multi-Word Proper Names with Morfeusz and Multiflex. LNAI **5070** (2009) 111–142
11. Saloni, Z., Gruszczyński, W., Woliński, M., Wołosz, R.: Słownik gramatyczny języka polskiego. Wiedza Powszechna, Warszawa (2007)
12. Krstev, C., Vitas, D.: An Effective Method for Developing a Comprehensive Morphological E-dictionary of Compounds. In: Proceedings of Lexis and Grammar Conference, Bergen. (2009) 204–212
13. Graliński, F., Savary, A., Czerepowicka, M., Makowiecki, F.: Computational Lexicography of Multi-Word Units: How Efficient Can It Be? In: Proceedings of the COLING-MWE'10 Workshop, Beijing, China. (2010) 1–9