An Authoring Tool for Controlled Modern Greek

Stella Markantonatou[1], Vangelis Karkaletsis[2], and Yanis Maistros[3]

[1]Institute for Language & Speech Processing
Epidavrou & Artemidos 6, Marousi, 15125 Athens, Greece
marks@ilsp.gr

[2]Software and Knowledge Engineering Laboratory
Institute of Informatics and Telecommunications, N.C.S.R. "Demokritos",
15310 Aghia Paraskevi Attikis, Athens, Greece
vangelis@iit.demokritos.gr

[3]Department of Electrical & Computer Engineering,
National Technical University of Athens, Athens, Greece
maistros@softlab.ece.ntua.gr

Abstract. We report on the first, to the best of our knowledge, attempt to define the core linguistic and formatting style specifications for controlled Modern Greek and develop an authoring tool (controlled language checker) in the context of the project "SCHEMATOPOIESIS". The tool is both parametric, in order to accommodate various thematic domains, and extensible, in order to host customised specifications at the level of (formatting) style, terminology and grammar. Two versions of the tool are presented, one operating within a word processing environment (MS-Word) and one operating on the Web. Both versions draw on the same linguistic and style specifications and share the same lexical and grammatical resources. The sublanguage of computational equipment has been used as a case study. The paper presents the design principles of the authoring tool, describes the two implementation versions and presents the first evaluation results as well as our plans for future research.

1 Introduction

Both humans and computers may experience difficulties in processing natural language due to its inherent ambiguity and complexity. Controlled languages handle this problem by ruling out troublesome structures and words. A controlled language is, by definition, a subset of natural language characterized by restricted syntax and vocabulary. Controlled languages are used when unambiguous texts are required; the case of technical documents is typical because a consistent technical writing style improves comprehensibility and adds to the quality of technical documentation. For example, a simple set of style guidelines for user documentation might be: *Make positive statements, Keep sentences short, Use only one idea per sentence, Use simple sentence structures, Use the active voice, Avoid conditional tenses, Use correct punctuation*. Such restrictions help to preserve uniformity in the writing style, especially in cases where authors tend to follow diverse writing approaches, and to reduce ambiguity in the resulting text.

The use of controlled languages facilitates translation, which strongly relies on a good understanding of the vocabulary, including terminology, and a fast disambiguation of the syntactic constructs used in the source text. Controlled languages reduce

ambiguity in the source text rendering the translation procedure more efficient and improving the quality of the output. The use of controlled languages also paves the way to machine translation systems because the resources already provided for controlled languages (vocabulary, terminology support and syntax rules) can be used for training a machine translation system. This reduces post-editing workload as well as the turnaround time for texts and the resources required for translation.

To the best of our knowledge, this paper presents the first attempt to define specifications for controlled Modern Greek and develop an authoring tool (controlled language checker). This effort was conducted in the context of the project "SCHEMATOPOIESIS[1]". The authoring tool presented in this paper relies on a number of linguistic and (formatting) style specifications. It has been designed to serve as a core system for controlled Modern Greek. The system is both parametric, in order to accommodate various domains, and extensible, in order to host customised specifications at the level of formatting style, terminology and grammar. Two implementation versions of the authoring tool are presented here, one operating within a word processing environment (MS-Word) and one operating on the Web. Both versions draw on the same linguistic and style specifications and share the lexical and grammatical resources developed in SCHEMATOPOIESIS. The sublanguage of the domain of computational equipment was used as case study.

In section 2 related work is presented, whereas in section 3 the design principles of the authoring tool are discussed. The two implementation versions are described in sections 4 and 5. In section 6 the evaluation results are presented while our plans for future research are presented in the concluding section 6.

2 Related Work

Controlled languages first appeared in 1930's when several linguists worked on the creation of a "subset" of English language, called 'Basic English' [1], that would facilitate the use of English by as many people as possible around the world. This approach was considerably different from previous efforts to create the so-called universal languages, since Basic English was a very well defined "subset" of an existing language (English) and not an artificial or hybrid language, such as Esperanto. One of the central ideas of Basic English was that the number of general-purpose words required for someone to produce texts ranging from a simple receipt to a speech on the world financial status, did not exceed a few hundred words compared to the approx. 75.000 words that are available [1]. This reduction of the length of the necessary vocabulary could be achieved by using "operator verbs" and a set of nouns or adjectives instead of verbs' derivatives which were normally preferred. For instance, the sentence *"The disc controller was perfected after several revisions",* was written in Basic English as *"The disc controller was made perfect..."*, where the verb *"to make"* is an "operator verb" and the adjective *"perfect"* is one of the

---

1 SCHEMATOPOIESIS is an R&D project funded partially by the Greek General Secretariat of Research & Technology (GSRT) and the EC. The project partners include Institute for Language & Speech Processing (coordinator), National Technical University of Athens, NCSR "Demokritos", ALTEC, UNISOFT.

allowed adjectives. The developers of Basic English recognized the need for extending the lexicon with the terminology of each thematic domain the controlled language applied to. However, even in the case that a document contained domain specific terminology, the use of the Basic English lexicon was enough to cover the general purpose words used in that document. Therefore, the conclusion was that for the production of technical documents no more was required than the Basic English lexicon and grammar rules together with the domain-specific terminology.

Controlled languages were initially used by large export industries in the USA. Instead of translating their technical documents in the languages of the countries they were exporting their products to, they decided to write them in a controlled English language. They assumed that technicians with a limited knowledge of English would easier read and understand texts characterised by simple syntactic structures and a simplified vocabulary. For instance, Boeing designed and used Boeing's Simplified English (BSE) in order to improve the readability of their technical documents [2]. BSE was based on Simplified English (SE), a standard defined for the air industry by the AECMA (Association Europeene des Constructeurs de Materiel Aerospatial). Caterpillar Tractor Company, USA also adopted a controlled language, called CTE (Caterpillar Technical English), to produce their technical documents.

As we mentioned in Section 1, controlled languages are useful not only for technical writing but also for translation either by human translators or by machine translation systems [12], [4]. Large software industries, such as Bull, Xerox, and Perkins Elmer are using controlled languages in combination with machine translation systems. Bull is using Bull's Global English (BGE) [3], which is based on the AECMA Simplified English. The documents produced with BGE are sent for translation to the machine translation system SYSTRAN, which is trained to use the grammar rules and vocabulary of BGE. A similar process is followed by Xerox which uses the Multinational Customised English (MCE) in combination with the machine translation system SYSTRAN and other translation tools. Perkins Elmer uses Perkins Approved Clear English (PACE) [1].

The work presented in this paper represents, to the best of our knowledge, the first effort to define linguistic and formatting style specifications for a controlled language for Modern Greek, and develop the appropriate authoring tool. For this reason, it drew a lot on similar work on other languages. However, it also took into account the linguistic and functional requirements of the potential users of such an authoring tool (i.e. the technical writers of the companies involved).

3 Design Principles

The design principles are the following:

- *Language level*: Reduction in ambiguity and redundancy – Effective terminology management [5], [6]
- *Formatting level*: Controlled text layout – The text layout reflects textual structuring
- *Implementation*: Use a development platform compatible to most current applications – Create a functional and user friendly tool

More specifically, at the language level, the effort is to eliminate ambiguity and redundancy (both lexical and structural) on the morphological, lexical (terminology included) and clause level (punctuation marks included). We have opted for robust approaches because we wanted to set the basics of Controlled Greek and to exploit the well-established linguistic technology which was available to the development sites in order to make sure that the development of the core system was a realistic task. All the resources developed and/or improved can be reused in future versions of the tool.

First of all, we made sure that the linguistic specifications comply with the international standards in the domain of language engineering (PAROLE, XML).

At the morphological level, we have tried to constrain the phenomenon of polytypia, which is prominent in Modern Greek. Polytypia exists when the same grammatical features correspond to more than one grammatical form. For instance, the set of grammatical features {Common Noun, Feminine, Singular, Genitive, NominativeSingularForm: *πόλη*} corresponds to two perfectly grammatical forms, namely, *πόλης* and *πόλεως*. The different grammatical forms often correspond to stylistic differences, which are inappropriate in a controlled language framework. Therefore, we have excluded certain classes of word forms of the nominal (1), verbal (2) and adverbial (3) paradigm. We have mostly relied on inflectional endings to identify these forms. In the case of adverbials, we have constructed lists of accepted adverbials which violate the morphological constraint in (3) for those adverbials in *–ως* which lexicalise a meaning other than their correspondent in *–α* (eg. ευχαρίστως (=with pleasure), ευχάριστα (=happily)) or have no correspondent in *–α* (eg. εκτενώς (=in length).

(1) reject: *-εως*, accept: *-ης* [πόλεως vs πόλης (= of the city)]

(2) reject: *-ουνε*, accept: *-ουν* [προσφέρουνε vs προσφέρουν (= they offer)]

(3) reject: *-ως*, accept: *-α* [απλώς vs απλά (= simply)]

At the lexical level, we forbid ambiguous words & phrases. More specifically, we have set constraints on several parts of speech, which have various functions, such as conjunctions (4) introducing several semantic types of subordinate clauses, prepositions with a multitude of meanings, pronouns, adverbs and interjections. In most cases one or more alternative words/phrases are offered (4a,b), (5a,b). By forbidding ambiguous functional words we prune some of the syntactic complexity of the language as function words introduce a variety of subordinated structures and, consequently, add to syntactic ambiguity.

(4) *Άμα* δείξετε πάνω στο εικονίδιο και διπλοπατήσετε, το εικονίδιο ανοίγει.

a. *Όταν* δείξετε πάνω στο εικονίδιο και διπλοπατήσετε, το εικονίδιο ανοίγει.
When you point to the icon and double-click, the icon opens.

b. *Εάν* δείξετε πάνω στο εικονίδιο και διπλοπατήσετε, το εικονίδιο ανοίγει.
If you point to the icon and double-click, the icon opens.

(5) *Πατώντας* στο κουμπί "Νέα διεύθυνση", μπορείτε να δημιουργήσετε μία νέα καταχώρηση στο βιβλίο διευθύνσεων.

a. *Όταν πατήσετε* στο κουμπί "Νέα διεύθυνση", μπορείτε να δημιουργήσετε μια νέα καταχώρηση στο βιβλίο διευθύνσεων.

When you click "New Address", you can make a new entry in the Address Book.

b. *Εάν πατήσετε* στο κουμπί "Νέα διεύθυνση", μπορείτε να δημιουργήσετε μια νέα καταχώρηση στο βιβλίο διευθύνσεων.

If you click "New Address", you can make a new entry in the Address Book.

The linguistic specifications also support an effective management of terminology. We have tried to control the use of terms in the text. As a case study we have taken the thematic domain of computer goods and have built an extensive database of approximately 3.500 multilingual terms (one and/or multiword terms as well as acronyms). We have imposed constraints on the way terms appear in the text and have used a checking mechanism, which crucially depends on the various fields of this database, in order to achieve successful term detection and recognition. According to this mechanism, a term which appears for the first time in a particular Greek text must be boldfaced and accompanied by its English translation, if it exists.

At the clause level, we have used surface grammar rules to eliminate the use of complex structures by forbidding specific configurations such as iterative phrase sequences, varied word ordering and continuous embedding. More particularly, as regards iterative phrase sequences, we have constrained the number of adjacent Noun Phrases in the genitive case, adjacent Prepositional Phrases as well as the number of prenominal adjectives. As regards word order, we have required that only two orderings are available, namely *Subject – Verb – Object* and *Verb – Subject – Object* while a main clause must always precede a subordinate one (6), expect for the case of temporal, causal, concessive and conditional clauses (7). Continuous embedding has been controlled by constraining the number of main verbal forms included in a period. The number of the available punctuation marks is also limited.

(6) Πατήστε στο κουμπί "Αποθήκευση", για να αποθηκεύσετε το έγγραφο.

Click "Save", in order to save the document.

(7) Όταν το παράθυρο είναι ελαχιστοποιημένο, η διαταγή "Μετακίνηση" είναι αδρανής.

When the window is minimised, the command "Move" is inactive.

At the formatting level, we have tried to establish a standard correspondence between textual structuring and the text layout. Our objective is to avoid ambiguity and vagueness not only with respect to language, but also with respect to text formatting. We have put effort in making the various kinds of text (titles, headers, captions, normal text, warning text etc.) easily discernible. We have created a formatting DTD (Document Type Definition), in which differentiating textual parameters such as font, font size, line spacing etc., are defined. These parameters help to easily distinguish among the various text types.

At the implementation level we followed two paths, one giving a word processor output (Microsoft Word was used in the current implementation) and the other an XML – HTML one browsable by any Web Browser. Both the implementations are robust, rely on the linguistic specifications presented and make good use of the technology available to the development sites. The overall result allows the user to access and exploit the core system in various environments.

4 Word Processor Based Version of the Authoring Tool

Technical writers can use the Authoring Tool through their word processor (Microsoft Word is used in the current implementation). Users can check the structure and language of his/her documents in a way similar to the one used with a spelling/syntax checker. The technical document is first converted into an XML format and is fed to the checker which outputs the error tags in a format "understandable" by the word-processor in order to let the user see his/her errors. The checker checks both text structure (e.g. line spacing, fonts style and size) and language (correct application of controlled language grammar and vocabulary) (see Fig. 1).

The XML text is processed using linguistic resources (restricted terminology, vocabulary, grammar) and tools in order to apply the language checker. More specifically, the linguistic processing tools are the following:

- Tokenizer: it identifies and characterizes tokens (e.g. a token type may be that the token is comprised of lower case Latin characters).
- Sentence splitter: it detects sentence boundaries.
- Part of speech tagger: this is a machine learning based tagger [11] that identifies part of speech and morphological features (gender, number, tense). The tagger output is according to PAROLE as specified in the controlled language.
- Case tagger: a machine learning based tagger that identifies the case for Greek nouns, adjectives and pronouns.
- Morphological analyser: it extracts from a morphological lexicon the morphological features for those words in the text for which a lexicon entry exists.
- Lexical analyzer: it combines the results of the taggers (part of speech and case tagger) with the results of the morphological analyzer in order to improve the results given to the lookup and checking modules following.

The Termbase/vocabulary lookup module locates those words, phrases or terms that exist in pre-stored lists (in our case the terminology and vocabulary lists). In order to reduce the lists size, we maintain only the lemmatised forms of the words included. For instance, there is one entry in the termbase for the term "τελικός χρήστης" (end-user) that covers the phrases "τελικός χρήστης" (nominative-singular), "τελικού χρήστη" (genitive-singular), "τελικό χρήστη" (accusative-singular), "τελικοί χρήστες" (nominative-plural), "τελικών χρηστών" (genitive-plural), "τελικούς χρήστες" (accusative-plural). This in turn requires the lemmatisation of the text, since the look up module attempts to match only the lemmatised forms. Lemmatisation is performed by the morphological analyzer during lexical preprocessing.

Language checking involves the following:

- *Punctuation marks checking*: locates the punctuation marks that are not included in the allowed list, or are not used according to the rules of the controlled language.
- *Part of speech checking*: checks the correct use (according to the rules of the controlled language) of pronouns, verbs, participles, etc.
- *Paragraphs, periods checking*: checks the paragraph size (in periods) and the period size (in sentences) according to the rules of the controlled language.

- *Titles, headings checking*: the existence of verbs or participles in titles and headings is prohibited by the rules of the controlled language.
- *Passive voice, genitive checking:* passive voice and consecutive nouns in genitive case are ruled out by the controlled language.
- *Terminology and vocabulary checking*: it locates words, phrases, terms and acronyms that are not allowed by the controlled language. In the case of terms and acronyms, it also checks whether the relevant style rules are met. Although this is part of the style checking, it is performed within the language checking module because it concerns linguistic data that cannot be located by the style checking module.

The formatting style checking is performed by a separate module according to the style specifications encoded in the format DTD of the controlled language. The DTD describes the allowed style tags, their attributes, their order as well as their allowed combinations in the text.

The linguistic resources and tools used have been developed using *Ellogon,* a new text engineering platform developed by NCSR "Demokritos" [11]. *Ellogon* provides a powerful infrastructure for managing, storing and exchanging textual data, embedding and managing text processing components as well as visualising textual data and their associated linguistic information. *Ellogon* was used not only as the development platform for the authoring tool, but also as *a means for embedding it into Microsoft Word*, allowing the user to check his/her documents in a similar way as a spell/syntax checker. All the components related to linguistic processing and language checking are running under *Ellogon*, whereas the components for generating an XML-based representation of the Word document, the components that perform the style checks and the components that mark the identified errors on the word document are running under MS Word. The communication between MS Word and *Ellogon* is achieved with the use of either ActiveX or DDE, both of which are services supported by both MS Word and the Windows version of *Ellogon*.

The language and style errors identified are presented to the user in a separate window (see Fig. 2). For some errors, the tool provides an indicative example in order to assist the user in the correction.

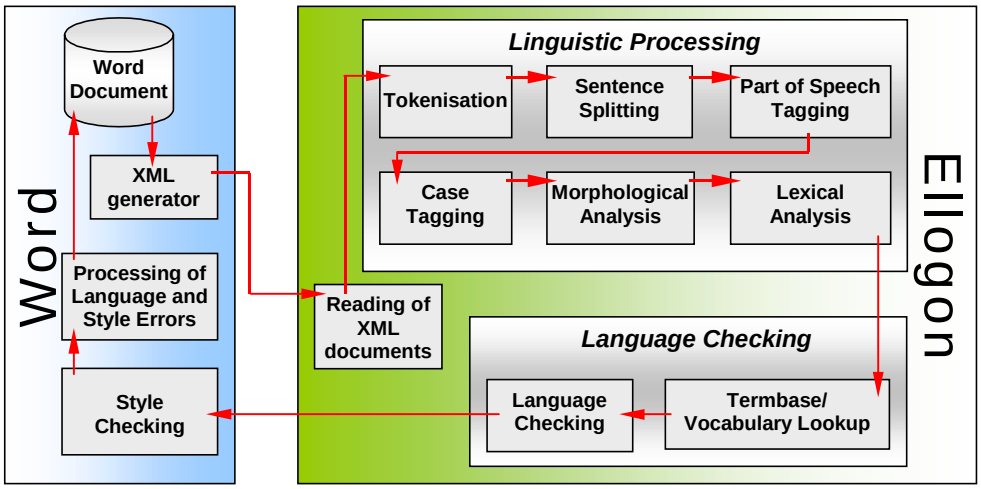

Fig. 1. The architecture of the word processor-based version of the authoring tool

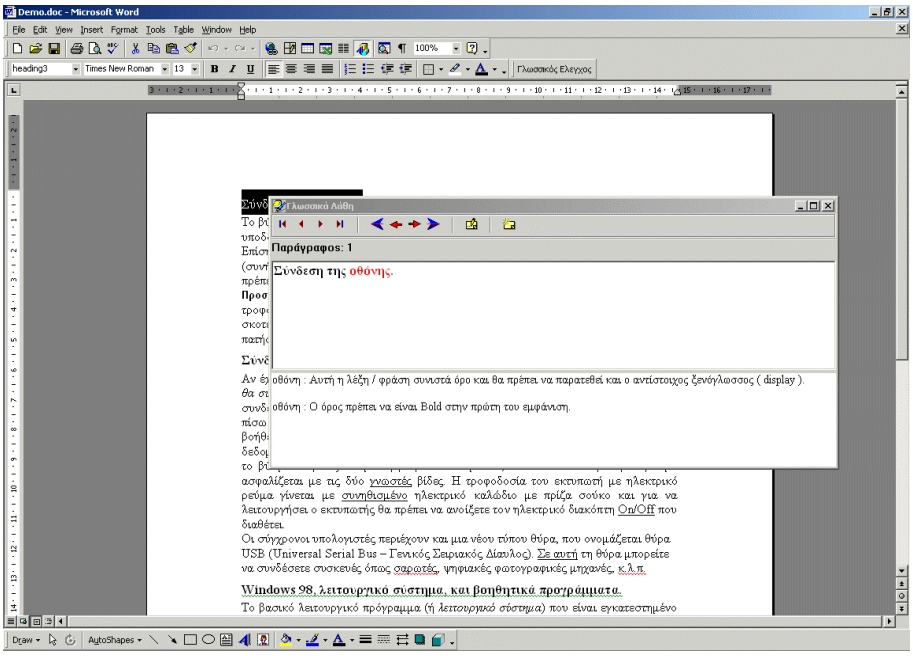

Fig. 2. Presentation of the results of the authoring tool

5 Web-based Version of the Authoring Tool

Web-based version of the authoring tool is running on a server to which the end user is connected. Users may submit to the tool their texts to be validated, by invoking the linguistic engine, a software system resident to the server. This engine triggers a client application which produces the final output after the check is accomplished.

Input to this version (see Fig 3.) can be any XML annotated document. The XML structure is assigned to the document either by the editor used to produce XML output (HTML-XML editors, emacs etc), or by an XML converter. XML annotation provides only style information.

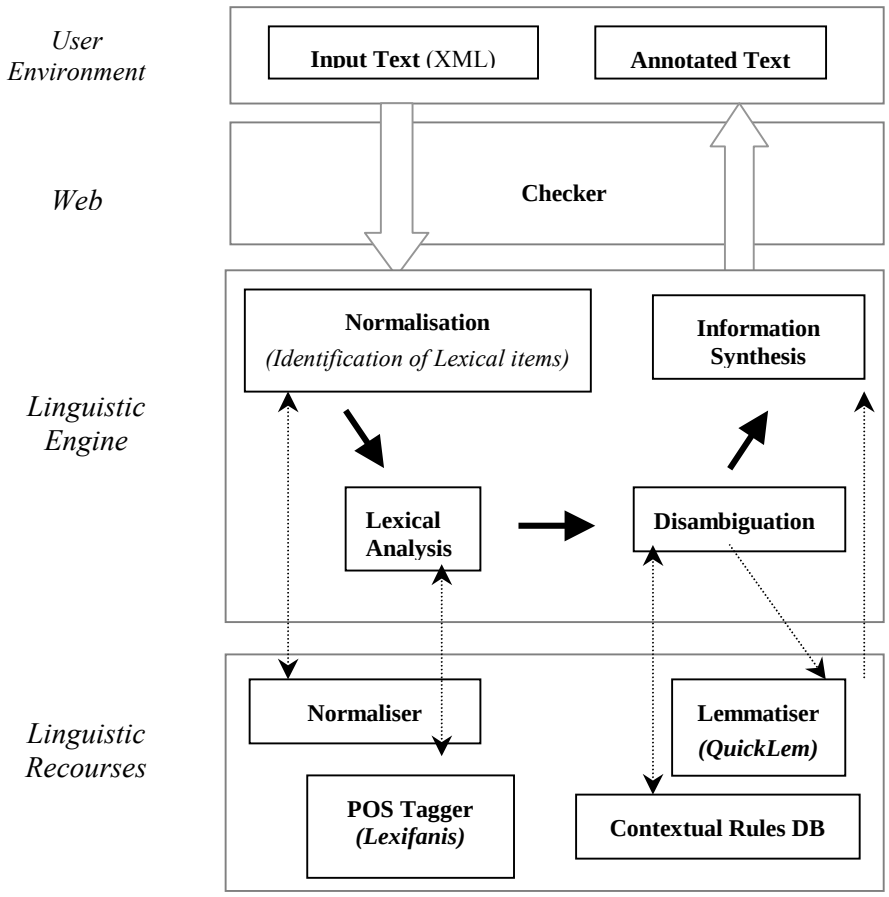

Fig. 3. The architecture of the web-based version of the authoring tool

The end user invokes the Web version of the Authoring Tool, supplies the system with his/her document and selects the group(s) of checks s/he wants the system to execute. The input text is first processed by the underlying linguistic engine, which

performs sentence splitting, tokenisation, Part-of-Speech tagging, grammatical annotation and lemmatisation. The obtained linguistic information is added to the existing XML structure in the form of PAROLE conformant tags.

Thus, the underlying linguistic engine performs the following distinct tasks while consulting a set of specialised linguistic resources:

- *Normalisation*: sentence splitting and tokenisation, performed by the Normaliser
- *Part-of-Speech and Grammatical Annotation*: performed by the "Lexifanis" POS Tagger [7]
- *Lemmatisation and Case Disambiguation*: carried out by "QuickLem" [8].

The "QuickLem" lemmatiser consults a database of inflectional endings and a limited set of contextual rules [9]. Contextual rules are used to resolve case ambiguity. None of the aforementioned tools makes use of a morphological lexicon. This is advantageous, because the overall application relies on "light" tools and a restricted amount of linguistic resources.

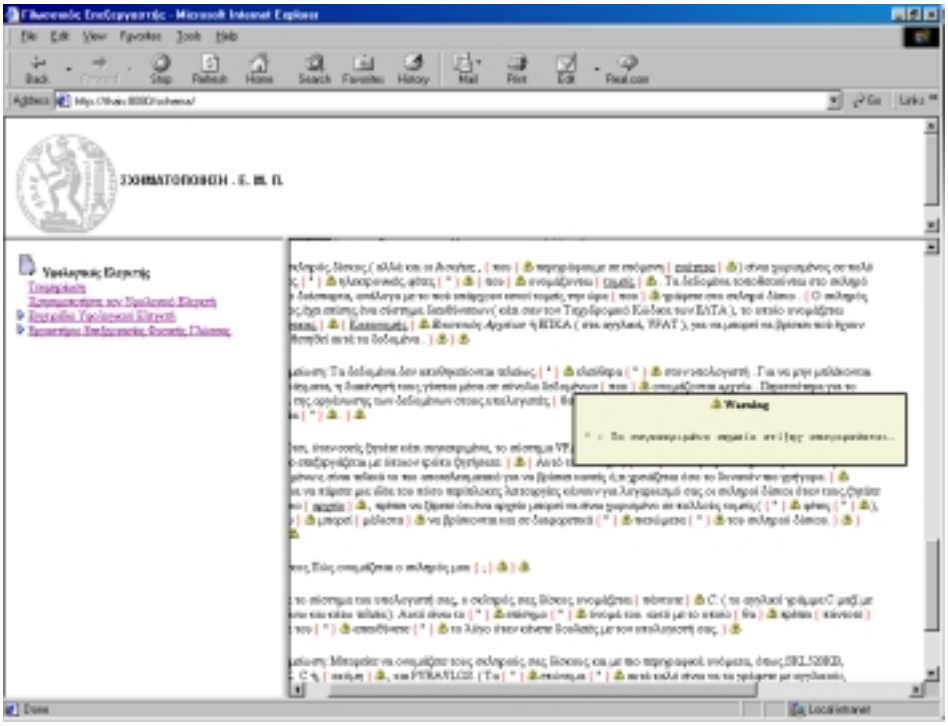

Fig. 4. Presentation of the results of the web-based authoring tool

The lemmatised and fully annotated XML text is then processed by the main checker module. Controlled language terminology and grammar rules are applied. If one or

more errors are detected in a text unit (word, phrase, sentence, paragraph), i.e. when one or more of the conditions imposed by the rules of the controlled language are not met, a warning message appears in a special window. The messages explain the errors and advise the user about possible corrections. This last module is written in Java and as a client based application, it incorporates all generated messages to the output produced (cf. Fig.4).

6 Evaluation Results

The two implementation versions were evaluated by two user groups. The first group included persons familiar with the functions of the authoring tool, whereas the second group included persons that knew nothing about the tool and its functions. Each user group was provided with the user and installation manual for both versions, the guidelines for writing in the controlled language, and a questionnaire. The users had to install the S/W (only in the case of the word processor based version) and then use the tool according to the user manual. Two types of document were processed. The first type consisted of documents containing specific linguistic and style errors. The second type included documents prepared by the users following the guidelines for writing in the controlled language.

The evaluation results have been quite positive. The users agree that both environments (MS Word and Web based) are user friendly and that this first prototype offers a useful aid to professional translators and language mediators generating technical documents. At this stage we have focused in evaluating the functionality of the system rather than comparing the two implementation versions. A contrastive evaluation will be necessary in the light of future expansion of the system into specific domains of application.

7 Concluding Remarks

In this paper we presented the core specifications for Controlled Modern Greek and the functionalities of the relevant authoring tool (controlled language checker) developed in the context of the project SCHEMATOPOIESIS. Because, to the best of our knowledge, this was the first effort of its type for the Modern Greek language, the system design drew on similar systems operating on other languages. We also took into account the linguistic and functional requirements of the potential Greek speaking users (i.e. the technical writers of the companies involved in SCHEMATOPOIESIS).

Special effort was put in creating a system that would be both parametric, in order to accommodate various domains, and extensible, in order to host customised specifications at the level of text style, terminology and grammar. For this purpose, we developed lexical resources (lexicons, terminologies, grammars) that can be easily re-used and adapted. We also adapted, according to the project needs, existing linguistic tools to take XML input while their output conforms to international standards for natural language processing (PAROLE).

Two versions of the authoring tool were developed, one operating within a word processing environment (MS-Word) and one operating on the Web. Both versions draw on the same linguistic and style specifications, share the lexical and grammatical resources and take the same input (i.e. the XML representation of the text).

We aim to further exploit the parametric and modular design of the system in order to extend its ability to handle technical documents in various domains.

References

1. Arnold, D., Balkan, L., Humphreys, R.L., Meijer, S., Sadler, L.: Machine Translation - An Introductory Guide. NCC Blackwell, Ltd (1994) 106-109, 147-164, 183-205
2. Boeing's Simplified English Checker. Language Industry Monitor - The World of Natural Language Computing, No. 13 (1993) 5-6
3. Bull Global English. Bull ILO, Paris (1993)
4. Eijk, P.: Controlled Languages in Technical Documentation. Elsnews, The Newsletter of the European Network in Language and Speech, February (1998) 4-5
5. Huijsen, W. O.: Controlled Language – An Introduction. CLAW'98 (1998) 1-15
6. Kamprth, C., and Adolphson, E.: Controlled Language for Multilingual Document Production: Experience with Caterpillar Technical English. CLAW'98 (1998) 51-61
7. Kotsanis, Y., and Maistros, Y.: "Lexifanis" - A Lexical Analyzer of Modern Greek. In: Proceedings of the 2nd Conference of the European Chapter of the ACL, Geneva (1985) 154-158
8. Kotsanis, Y., Maistros, Y., and Zavras, A.: "Quicklem" - A Software System for Greek Word-Class Determination, Literary and Linguistic Computing, Vol. 2, No 4, Oxford University Press (1987) 242-244
9. Maistros, Y., Vassiliou, M., and Markantonatou, S.: QUICKLEM: How you find the lemma without a morphological lexicon. To be included in: Proceedings of the 5th International Conference in Greek Linguistics, Université Paris 5, Sorbonne, 13-15 September (2001)
10. Petasis, G., Karkaletsis, V., Paliouras, G., and Spyropoulos, C.D.: Ellogon: A Text Engineering Platform. NCSR "Demokritos", Technical report (2001)
11. Petasis, G., Paliouras, G., Karkaletsis, V., Spyropoulos, C.D. and Androutsopoulos I.: Using Machine Learning Techniques for Part-of-Speech Tagging in the Greek Language. In: Proceedings of the 7th Panhellenic Conference on Informatics, Ioannina, Greece, August (1999)
12. Vouros, G., Karkaletsis, V., and Spyropoulos, C.D.: Documentation and Translation. In: Hall, P.A., and Hudson, R. (eds.): Software without frontiers. J.Wileys & Sons (1997) 167-202