UDC 81'367

DEEP PARSING FOR THE AVIATION INDUSTRY: ADJUSTING ARTEMIS FOR PARSING SIMPLE CLAUSES IN ASD STE-100*

A. Díaz-Galán
Universidad de La Laguna, Instituto de Lingüística Andrés Bello (Tenerife, Spain)
adiazgal@ull.edu.es

The linguistically motivated parser ARTEMIS (Automatically Representing Text Meaning via an Interlingua-based System) has been designed [Periñán-Pascual and Arcas Túnez 2014] with the aim of providing formal representations of natural language fragments enriched with syntactic, semantic and pragmatic information. To test the validity of a parsing system before it is applied to natural language input it is a standard practice to apply it to a controlled language. In this paper we will address the treatment of simple clauses in ASD-STE 100, the controlled natural language employed by the European aviation industry for technical documentation.

We will create or revise lexical and syntactic rules within ARTEMIS in order to make them sensitive to the formal impositions and the restricted communicative functions allowed in this technical language. These rules -which are consistent with the tenets of two linguistic models which substantiate ARTEMIS -namely, Role and Reference Grammar and the Lexical Constructional Model- will account for the first time for each of the different nodes that describe simple sentences in ARTEMIS, in attempt to make it suitable for the parsing of ASD-STE 100.

We will conclude by indicating the areas where further research is needed for the full implementation of the parser.

Key words: *Functional Grammar Knowledge Base (FunGramKB), ARTEMIS, Natural Language Processing (NLP), ASD-STE100 Controlled Language, sentence parsing, computational application of RRG.*

For citation: Díaz-Galán, A. (2018). Deep Parsing for the aviation industry: Adjusting ARTEMIS for parsing simple clauses in ASD STE-100. *Voprosy Kognitivnoy Lingvistiki*, *3,* 83-96.

DOI: 10.20916/1812-3228-2018-3-83-96

Introduction

In an era where social interactions, commercial exchanges and labour are increasingly dependent on computer mediation, there is – as has always been the case for exchanges between speakers of different languages – an equally large surge in the demand for translation. This new type of interaction presents one outstanding and obvious characteristic, namely, that one of the participants is not a natural language user but a processor that can only read code to execute orders. To bridge the gap between the two, Natural Language Processing (NLP) employs parsers. These programs are basically algorithms – the interpreters- which aim at transducing the natural text input into a formalized output representation that, ideally, should be able to convey the depth and richness of human language. This is not, however, the ultimate purpose of most NLP parsers, since many are conceived as mere tools for representing the formal structure of the input text. Driven by the immediacy of the human demands for technological developments and by the requirements of the industry, many parsers tend to neglect language nuances for the sake of effectiveness in carrying out tasks such as automatic summarization, machine translation, sentiment analysis, question answering, tagging, text classification, data mining, or, in the case of the aviation industry, translating natural language instructions and procedures into an unambiguous code that could allow robots to perform complex maintenance tasks which, so far, can only be carried out by human beings.

In this context, it does not come as a surprise that the most employed parsers rely on "shallow" parsing, a procedure, which, initially, would only group elements into syntactic constituents employing the formalisms of context free grammars. When these

* **Acknowledgements:** This work has been developed within the framework of the research project "Desarrollo de un laboratorio virtual para el procesamientocomputacional de la lenguadesde un paradigmafuncional". (UNED) FF2014-53788-C3-1-P funded by the Spanish Ministry of Education, Culture and Sports.

parsers use for their analyses statistical information derived from the regularities of corpora they are termed probabilistic parsers. The latter make inferences about the structures of the samples of natural language that they analyse, discarding or accepting options on the basis of their statistical likelihood, a procedure that speeds up the parsing process considerably. Shallow parsing -which proves efficient for certain tasks which require fast disambiguation such as data mining or as a previous step for further processing-is not enough, however, if the final aim to transduce not just the syntactic form, but the nuances which characterize the written or oral natural language input is necessary, for instance, for dialog systems. Such complex undertakings call for a deeper approach to parsing; an approach that can enrich the straightforward syntactic analysis of context free grammars and grant access to the semantic, pragmatic or even discursive information concerning the different units processed. To this end, parsers can be supplied with Attribute Value Matrixes (AVMs), formalized feature bearing devices which allow a thorough linguistic description of any constituent. The specifications contained in the AVMs are merged or "unified" with the rules for the different syntactic constituents, imposing restrictions and constraints on them, which results into a finer grained processing. This type of approach is what characterizes a range of grammars frequently employed in deep parsing known as Unification Grammars [eg. Sag, Wasow and Bender 2003].

ARTEMIS (Automatically Representing Text Meaning via an Interlingua-based System) can be described as a deep parser which performs unification processes. It has been designed [Periñán-Pascual and Arcas Túnez 2014] to provide a formal representation of natural language fragments and, when completed, will not only be able to create syntactic trees, but it will also enrich the representation with semantic roles, aspectual information *(Aktionsart)* and constructional meaning. The ultimate purpose of the parser is to achieve formal representations as the one provided in (1), where a natural language sentence is eventually transduced into a rich rendering that employs the language formalisms of FunGramKB [Periñán-Pascual and Arcas Túnez 2007], the environment to which ARTEMIS belongs.

(1) *Louise baked a cake for the kids.*

COREL schema: + (e1: +BAKE_00 (x1: %LOUISE_00)THEME (x2: +CAKE_00)REFERENT (f1: (e2: +DO_00 (x1)AGENT (e1)REFERENT (f2: +CHILD_00)Beneficiary))Purpose) [Fumero-Pérez and Díaz-Galán 2017: 38]

The fact that the formal representations aimed at in ARTEMIS can tag semantic and constructional information clearly indicates that the parser has been created with a solid linguistic motivation. The two functional linguistic theories which frame the design of ARTEMIS are the Lexical Constructional Model [LCM henceforth; Ruiz de Mendoza and Mairal-Usón 2008] and Role and Reference Grammar [RRG henceforth; Van Valin and La Polla 1997, Van Valin 2005]. The LCM provides a rich description of non-compositional or constructional meaning which encompasses four different levels: L1 or argumental; L2 or implicational; L3 or illocutionary and L4 or discursive. Complementary to the LCM, Role and Reference Grammar has a typologically adequate grammatical description of the language that is sensitive to semantics and also amenable to formalization [Cortés-Rodríguez and Mairal-Usón2016]. Syntactic clausal analysis in ARTEMIS is based on the Layered Structure of the Clause (LSC) postulated by RRG. This theory distinguishes an operator projection where abstract grammatical information (eg. tense, mood, aspect, negation) is provided and a constituent projection that specifies the different syntactic nodes (Predicate, Nucleus, Pre-Core, Core, Clause, Sentence and Left Detached or Right detached Positions) and their realizations. Applying RRG in a computational environment has required two main adjustments [Cortés-Rodríguez 2016, Mairal-Usón and Cortés 2017]: a) merging the information contained in the operator projection with the constituent projection by means of AVMs and unification processes, b) modifying the LSC to include a specific node sensitive to argumental constructions, the CONSTR-L1 node (figure 1).

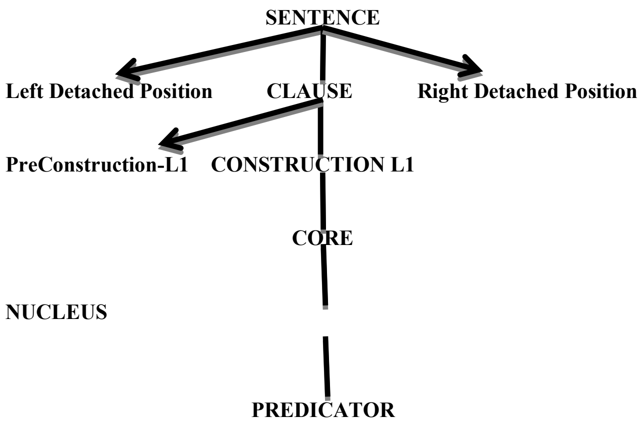

Figure 1. Modified Layered Structure of the Clause in ARTEMIS

Achieving the type of deep parsing that ARTEMIS pursues is not a straightforward task, as linguists will first have to supply the program with all the necessary information to complete the description of the units parsed. ARTEMIS, for instance, has access to conceptual meaning via the FunGramKB Ontology, it is also connected to a Lexicon which provides lexical and morphosyntactic information, and to a Grammaticon which stores constructional information. These three elements, which also belong to the FunGramKB environment, had to be fully operative before ARTEMIS could be implemented. Developing the ARTEMIS application itself implies storing in a dedicated component – the Grammar Development environment (GDE) – the lexical, syntactic and constructional rules that account for the different realizational possibilities of the units, as well as completing the information needed in the corresponding AVMs. This meticulous procedure aimed at achieving a fine-grained processing is, however, time consuming, and can result into what is frequently termed a "parsing bottleneck".

A compromise solution to speed up the design of the parser is to restrict its scope from the ultimate aim of processing full natural language to parsing a controlled natural language [henceforth CNL; Schwitter 2010, Khun 2014]. CNLs are natural languages which have been manipulated to create simplified codes restricted in lexis, syntax and, sometimes, communicative functions [Khun 2014: 123]. If the specific purpose of the CNL is to prevent ambiguities when non-native speakers read technical documents written in a foreign language, they are termed Controlled Technical Languages. These specialized languages are based on a natural language – most frequently English – but employ only "a well-defined subset of a language's grammar and lexicon" [Kittredge 2003:441, via Khun 2014:122]. In the case of ARTEMIS, we are using the controlled technical language ASD STE-100 to test the validity of the parsing system before it is applied to natural language input.

In the remainder of this paper we will revise the treatment of simple clauses in ASD STE-100 with the aim of allowing ARTEMIS to parse simple clauses in this technical language. This will involve adapting the syntactic rules and the AVMs stored in the GDE within ARTEMIS or creating new ones to account for the restricted communicative functions allowed in the language. Accordingly, in the following sections we will summarize the language characteristics of ASD STE-100 in relation to the sentence (section 2) and discuss the main areas where adjustments to the parser are required (section 3). These, in our opinion, are the treatment of relevant function words (section 4), and the modification or design of the rules that describe the different nodes of the LSC (section 5). We will conclude summarizing the contribution of this paper toward the development of the ARTEMIS parser and indicating the areas that deserve future research.

1. Sentence treatment in the natural controlled technical language ASD STE-100

ASD STE-100 (STE henceforth) is an English-based simplified technical language employed as the standard for written documentation concerning aviation maintenance by the ASD, the association which groups European aeronautics, space, defence and security industries. To grant the standardization and disambiguation of aviation maintenance documentation, these texts have to comply with the guidelines stated in the STE specification manual [2017]. Such manual consists of a list of writing rules and a restricted natural language dictionary that can be complemented with the technical terminology specific to a particular manufacturer. The dictionary compiles what they call "approved words", together with their "approved forms" and meanings, and provides alternatives for "not approved" words. The section devoted to the writing rules spells out a range of regulations that affect lexical, morphological, syntactic or textual phenomena. Some of these specifications are especially relevant for parsing simple sentences within ARTEMIS, since they will affect both the syntactic rules and the AVMs. In our opinion, the main ones are the restrictions concerning verbs and verb phrases and the formalization of the communicative functions which characterize instructions manuals, namely, describing procedures and giving instructions [Sharpe 2014, Mayes 2003].

There are a number of rules that concern verbs in STE, the most relevant for our purposes are the restrictions on the use of participial forms and auxiliaries. In STE, *-ed* and *–ing* participles can only be used as participial adjectives, and, at the same time, auxiliaries cannot be used to form what they call "complex structures". These indications, *de facto*, rule out passive voice and aspectual distinctions, and impose a severe reduction on the set of primary and modal auxiliaries. In fact, in STE primary auxiliaries are limited to *do*, which will only function as an auxiliary of negation or of emphasis. As for modal auxiliaries, the allowed forms are *can, cannot, could* and *must*. These will present mainly deontic modal meanings, although *can* and *could* are also employed to convey possibility. Finally, a further reduction on the auxiliaries is brought by the prescription against the use of contracted forms, which discards any form of enclitic auxiliaries.

The discursive nature of this type of instructional texts entails that we will find two main rhetorical functions: instructing and explaining. Instructions as a general category can be defined as directives that explain how to perform a procedure [Mayes 2003]. As subordinate communicative functions, instructions can also express cautions or warnings (in fact, negative instructions can be considered warnings). The specification states that instructions have to be expressed as short (less than 20 words) and unambiguous imperative sentences. These, however, can also be formulated by means of declarative sentences which contain the deontic modals allowed, as in (2d). The following are examples of the different types of instructions from the Airbus corpus[1]:

(2) a. *Disconnect the hydraulic pump* (procedure)
b. *Be careful when you use consumable materials* (caution)
c. *Do not breathe the fumes* (warning)
d. *You must remove unwanted hydraulic fluid immediately* (procedure)

Instructions can be preceded by preconditions, this is, actions, states or events that specify the conditions of the instruction. The manual states that they should appear at the beginning of sentence separated by a comma. Common examples display initial adverbial clauses:

(3) *When you inflate the tire, make sure that the pressure is not more than the maximum limit*.

The other main type of rhetorical function we can find in instruction manuals is providing explanatory information. This information is coded as short statements (25 words max.) which can be grouped in paragraphs of no more than six lines. Syntactically speaking, they are positive or negative declaratives with various degrees of structural complexity. Often, they also present a precondition, as in (7):

(4) *Each strut has two wheels in twin configuration.*

(5) *A hydraulic actuator opens and closes each door*.

(6) *Nose wheels stay at a fixed position because the axle nut prevents axial movements.*

(7) *Before a new autobrake mode engages, the active selection disengages*.

As can be seen in the following examples, when these sentences contain modal auxiliary verbs they also convey the values associated with deontic or epistemic modality. Notice, as well, the emphatic value of the primary auxiliary *do* in (10):

(8) *The area around the aircraft must be clear of Persons, Access platforms, Tools and equipment* (obligation)

(9) *You can use the Captain's tiller* (permission)

(10) *Damage to the hose(s) can occur if they do twist* (possibility/ emphasis)

As we can gather, the language associated with instructional genres and the writing prescriptions stated in the specification manual for STE entail that sentences will be restricted to the declarative or imperative form. Seeking to convey the content in the most unambiguous and straightforward manner possible, also derived from the nature of instructions manuals, we will not find questions or pragmatically marked phenomena other than emphatic *do*. This however, does not imply that sentences will be equally simple at structural level; to the contrary, they can present fairly complex structures due to the co-occurrence of adjuncts (11) or to the recursive use of coordination and subordination (12):

(11) *They are started manually from the aircraft integrated monitoring and diagnostic system home page on the aircraft multipurpose aircraft access terminal*.

(12) *These chambers contain part of the restrictors and flow controls to control the extension and retraction rates and gives a smoother range of travel*.

2. Accounting for STE sentences in ARTEMIS

To date, research on the parsing of simple sentences within ARTEMIS has succeeded in producing partial rules for declarative sentences [Mairal-Usón and Cortés-Rodríguez 2017]; for *yes/no* and *wh*-questions [Martín-Díaz 2017]; for sentential negation realized by enclitic auxiliaries [Díaz-Galán and

[1] Unless indicated otherwise, the examples belong to a collected corpus of texts from aircraft maintenance instructions, courtesy of Airbus, Seville.

Fumero-Pérez 2016] and, finally, for the integration of argumental constructions in the structure of sentences [Fumero-Pérez and Díaz-Galán 2017]. In the light of the characteristics we have just reviewed, parsing clausal units in STE will require, on the one hand, developing rules for syntactic phenomena not yet described and, on the other, modifying or discarding existing rules and AVMs. If we want ARTEMIS to be able to carry out an analysis of clauses for this language, it will have to accommodate its parsing routine to the following characteristics of sentences:

1. Formally sentences can only be declaratives or imperatives. There are no interrogatives.
2. Negation is formed by combining *auxiliary verb* + *not*. It is never enclitic with the exception of "*cannot*".
3. There are no pragmatically marked constituents except for emphatic *do*.
4. There is hypotaxis and parataxis.
5. There is a frequent co-occurrence of adjuncts.

In what follows we will concentrate on the first three as a way to develop the capacity of ARTEMIS to parse simple sentences and the most common syntactic phenomena in this controlled language. The rules stated in the manual that restrict the different types of clauses and the possible operations on them have an effect on two major areas:

a) The inventory of function words in ARTEMIS will have to be refined as some parts of speech (POS) will disappear and some others will be restricted in their attributes and values.

b) The description of the different syntactic nodes which make up the layered structure of the clause will have to be redefined, a process which will involve modifying AVMs and syntactic rules.

3. Adjusting POS

POS comprise a set of closed class of function words such as pronouns, conjunctions or auxiliaries, which are stored together with their description in the GDE. This description is formalized as a lexical rule and created in runtime by the program gathering the information that linguists have previously provided in the form of AVMs. It is very relevant, then, that the information contained in the AVMs for POS is accurate for the parser to be able to create the lexical rule. In the specific case of STE, the catalogue of function words within ARTEMIS has to be adjusted either by reducing elements, or by adding new ones which had not been contemplated in previous versions. In our opinion the most relevant ones are those affecting auxiliary verbs and accounting for negatives with *not.*

3.1. Modifying primary auxiliaries

Since in STE there is no passive voice and aspectual distinctions between perfect and continuous tenses are blocked, the set of primary auxiliaries is drastically reduced to the single auxiliary *do*. Different from natural language use, this primary auxiliary cannot be used for yes/no interrogatives or for negation (enclitic *doesn't, don´t* and *didn´t* are not allowed) but it can, however, have an emphatic value. The redefinition of *do* implied that we had to modify the category AUX originally intended to account for primary auxiliaries. In ARTEMIS these auxiliaries are characterized [Martín-Díaz 2017] in terms of Aspect (perfective or progressive); Illocutionary force (declarative, interrogative or imperative); Number (*Num*); Person (*Per*); Tense (present, past and future or null)[1] , and the syntactic restrictions imposed on the following verbs (*Syn*)[2], In STE, however, aspect, illocutionary force and polarity have been made redundant:

(13) AVM for STE AUX(*do,* positive):

```
<Category Type="AUX">
   <Attribute ID="Emph"/>
   <Attribute ID="Num"/>
   <Attribute ID="Per "/>
   <Attribute ID="Syn"/>
<Attribute ID="Tense"/>
```

Notice, as well, that emphasis (*Emph*) has been added to the list of attributes. To be able to account for the emphatic use of *do,* we have created the tag *Emph*, to which we have assigned the values emphatic (*e*) or non-applicable (*null)*:

---

[1] In RRG, Tense is a clausal operator which "situates the proposition expressed by the clause within the temporal and realis-irrealis continua" [Van Valin 2005:9]. In ARTEMIS, as a result of the unification processes [Cortés-Rodríguez and Mairal-Usón 2016], the tense values will percolate from the Nucleus to the Clause node. Initially the values assigned to this operator were present, past or null, while future values were expressed via the epistemic auxiliary *will* [Martín-Díaz 2017]. As a tentative solution for the absence of epistemic meanings other than possibility in STE-100, we have analysed *will* with a future value as one of the realizations of the operator Tense.

[2] As complex verb forms and passive voice are ruled out in STE-100, the attribute syntactic agreement *(Syn*) can only have the value (bare) *verb*.

(14) AVM for EMPH:

```
<Attribute ID="Emph" obl="*" num="1">
    <Value Tag="emph">e</Value>
    <Value Tag="never">null </Value>
</Attribute>
```

The resulting lexical rule for AUX in STE is spelled out in (15), while (16) shows how the attributes of emphatic *do* would be saturated in a clause:

(15) Lexical rule for AUX (*do*) in STE:

AUX [emph= e | null,num= pl | sg | null, per= 1 | 2 | 3, syn= verb | null, tense= pres | past | fut | null]

(16) *They do twist.* *do*: [emph= e , num= pl, per= 3, syn= verb, t= pres]

3.2. Modifying modal auxiliaries.

In RRG, deontic and epistemic modality are associated with different operators and have different scopes: deontic modals correspond to the operator Modality and affect the Core node, while epistemic modals are related to the operator Status and influence the Sentence. In ARTEMIS, these differences have been accounted for by creating the tags MODD for deontic modals and MODST for epistemic modals. They have been described in AVMs [Cortés-Rodríguez and Mairal-Usón 2016], and inserted in the rules for the Core and Sentence nodes [Martín-Díaz 2017]. To suit its rhetorical purposes, in STE deontic modality only comprises the forms *can, cannot* and *could* when they convey ability or permission, and *must* when it expresses obligation. Epistemic modality is realized by the auxiliaries *can* and *could* when they are associated with possibility, as in (10). Accordingly, the descriptor of both categories, MODD and MODST, had to be adjusted.

As was the case for *do*, deontic auxiliaries in STE cannot mark changes in illocutionary force. They can, however, present different modality (*Mod*) values (*abl*: ability, *obl:* obligation, *perm*: permission), tense distinctions, and polarity. These characteristics are described in (17):

(17) AVM for ASD STE MODD: *Deontic can, cannot, could, must (positive)*

```
<Category Type="MODD ">
    <Attribute ID="Mod"/>
    <Attribute ID="Pol"/>
    <Attribute ID="Syn"/>
    <Attribute ID="Tense"/>
</Category>
```

In (18) we spell out the lexical rule for MODD in STE with all the range of values which can saturate each of the attributes. These features are meant to account for the different uses of *can, could,* and *cannot* as deontic modals, as the lexical rules in (19) summarize:

(18) MODD [mod= abl | obl | perm, pol: pos | neg, syn= verb | null, tense= pres | past | fut | null]

(19) *can*: [mod: abl | perm, pol: pos, syn= verb, tns: pres]

could: [mod: abl | perm, pol: pos, syn= verb, tns: past]

cannot: [mod: abl | perm, pol: neg, syn= verb, tns: pres]

The AVM for epistemic modality in STE (20) is also considerably shortened. MODST only features attributes for Status (*Sta*)- that can only be saturated with the value possibility (*poss*)-, for syntactic agreement, and for tense. Notice that, as the only epistemic modals allowed are in the positive form, the polarity attribute is not necessary.

(20) AVM for STE MODST: *Epistemic can, could (positive)*

```
<Category Type="MODST ">
<Attribute ID="Sta"/>
<Attribute ID="Syn"/>
<Attribute ID="Tense"/>
</Category>
```

The following lexical rules summarize the attributes and values of the revised category MODST and illustrate the values of epistemic modals in STE:

(21) STE MODST[mod= poss, syn= verb | null, tense= pres | past | fut | null]

(22) *can:* [sta: poss, syn= verb, tns: pres]
could: [sta: poss, syn= verb, tns: past]

3.3. Clause Negation

Within ARTEMIS, clause negation had only been dealt with in cases where it was carried out by contracted primary or modal auxiliaries [Díaz-Galán and Fumero-Pérez 2016, Martín-Díaz 2017]; since in STE enclitics are not allowed, we had to devise the rules to account for clausal negation with *not*. A decision had to be made as to whether we should treat this element as an adverb, in line with traditional approaches to grammar [eg. Quirk et al. 1985], or if, for parsing purposes, it was more convenient to analyse it as a function word. The complexity of the analysis of adverbials tipped the balance towards the compromise solution of parsing them as a POS, as this would allow us to include them in the rules for negative clauses before tackling the problematic issue of parsing adverbials. To account for the word *not*, both in ARTEMIS and in its version for STE, we created the label NEG and its corresponding AVM. It only contains the attribute“assertion” (*Assr*), and this, in its turn, can present the values assertive (*assr*), negative (*neg*) or non-assertive (*noa*). This formula allows the AVM for NEG to be equally valid to account for other function words (i.e. quantifiers).

(23) AVM for NEG:

```
<Category Type=“NEG">
<Attribute ID="assr/>
</Category>
```

(24) AVM for ASSERTION:

```
<Attribute ID=“Assr" obl="*" num="1">
    <Value Tag=“assr">as</Value>
    <Value Tag="neg">neg</Value>
    <Value Tag="noa">noa</Value>
</Attribute>
```

(25) *not*: [assr: neg]

4. Adjusting the syntactic nodes of the LSC in STE

Parsing simple declarative and imperative sentences in the positive or in the negative within STE requires the revision of the AVMs and the syntactic rules which describe each of the nodes of the LSC in ARTEMIS. The information contained in these AVMs will be merged with the rules, resulting in an enhanced LSC. In this section we review the most important modifications that the different levels have undergone; we revisit their AVMs and provide – for the sake of readability – a simplified version of the syntactic rules that we list in their full version in appendix I.

4.1. The Nucleus

The first of the nodes of the LSC that needs a reformulation is the Nucleus (NUC), as it is at this level that we locate the operators for emphasis, and, in line with RRG, for aspect and negation [Van Valin 2005: 9]. To our purposes, the status of sentential negation with *not* in RRG is especially relevant, for it could be a feature pertaining to the NUC or to the Core [cf. Díaz-Galán and Fumero-Pérez 2016]. An argument in favour of considering it as part of the NUC is that, in natural language, when *not* is combined with aspectual auxiliaries (eg. ${}_{\text{NUC}}$[*have* $_{\text{aspect}}$ *done*]) it will necessarily appear after the nuclear operator *aspect* (eg. ${}_{\text{NUC}}$[*have*$_{\text{aspect}}$*not*$_{\text{neg}}$ *done*]), which seems to indicate that *not* also belongs to the NUC node. Since in STE aspectual variation is not permitted, the AVM for the NUC (26b) will not feature the aspect attribute, but, it will include, however, the attribute *Negation (Neg):*

(26) AVM for STE NUC:

```
<Category Type="NUC">
    <Attribute ID="Concept"/>
    <Attribute ID =“Emph"/>
    <Attribute ID="Illoc" />
    <Attribute ID="Mod" />
    <Attribute ID="Num" />
    <Attribute ID="Neg" />
    <Attribute ID="Per" />
    <Attribute ID="Recip" />
    <Attribute ID="Reflex" />
    <Attribute ID="Sta" />
    <Attribute ID="Template" />
    <Attribute ID="Tense" />
</Category>
```

(27) STE NUC [concept: ?, emph: ?, illoc:?, mod: ?, neg: ?, num:?, per:?, recip:?, reflex:?, sta=? tpl:?, t: ?] → PRED[concept= ?, illoc=?, num=?, per=?, sta= ? tpl=?, t=?] ||

The arrangement of the different constituents of the NUC node within ARTEMIS was spelled out by a syntactic rule devised by Cortés-Rodríguez and Mairal-Usón [2016: 100] which contemplated twelve different syntactic patterns. The equivalent rule in STE (28) is reduced to half size, mainly because of the lack of aspectual distinctions and the inclusion of the new nuclear operators *Emph* and *Neg*.

(28) Rule for NUC in ASD STE (simplified)

NUC → PRED: twist/twists/twisted
NUC → AUX PRED: do twist
NUC → AUX NEG PRED: do/did not twist
NUC → MODD PRED: can/could/must/cannot twist
NUC → MODD NEG PRED: can/could/must not twist
NUC → MODST PRED: can/could/occur

4.2. The Core

The Core, the node which comprises the NUC and its arguments, is the level at which we can account for declarative, interrogative and imperative structures. The lack of interrogatives in STE imposes, therefore, a reduction of the syntactic rules; while, at the same time, in the AVM new attributes will percolate from the NUC to higher levels via unification. Accordingly, the AVM for the Core in ARTEMIS [Cortés-Rodríguez [2016: 81] was updated for STE as follows:

(29) AVM for STE CORE:

```
<Category Type="CORE">
    <Attribute ID="Concept" />
    <Attribute ID= "Emph"/>
    <Attribute ID="Illoc" />
    <Attribute ID="Mod" />
    <Attribute ID="Neg" />
    <Attribute ID="Per"/>
    <Attribute ID="Recip" />
    <Attribute ID="Reflex" />
    <Attribute ID="Sta" />
    <Attribute ID="Template"/>
    <Attribute ID="Tense" />
</Category>
```

(30) STECORE [concept=?, emph: ?, illoc=?, mod=?, neg=?, num=?, per=?, recip=?, reflex=?,sta=?,tpl=?, t=?]

The syntactic rule for the Core in declarative sentences formulated by Mairal-Usón and Cortés-Rodríguez [2017] for ARTEMIS also required certain amendments, since the controlled language declaratives can be emphatic or present clause negation with *not*. Adjusting the attributes of syntactic rule for the Core in declaratives (see appendix 1) allows us to account for these realizational possibilities as variants of the three basic types of verb complementation:

(31a) CORE → ARG NUC (*Kernel 1*):
The engine stops/-ed/ DOES/DID NOT stop
The engine DOES stop.
The engine can/could/must/NOT/ stop.
The engine cannot stop.

STECORE [concept=?, emph: ?, illoc=DEC, mod=?, neg=?, num=?, per=?, recip=?, reflex=?, sta=?, tpl=?, t=?]->ARG[concept=?,macro= a | u | n, num=?, per=?, phrase=?, role: agent | attribute | goal | instrument | location | manner | origin | referent | result | theme, tpl=?, var= x | y | w | z] NUC [concept: ?, emph: ?, illoc:?, mod: ?, neg: ?, num:?, per:?,recip:?, reflex:?, tpl:?, sta=?t: ?]

(31b) CORE → ARG NUC ARG (*Kernel 2*):
The cylinder absorbs the impact /It does NOT absorb ...
The cylinder DOES absorb ...
The cylinder can/could/must/NOT/ absorb ...
The cylinder cannot absorb the impact.

STECORE [concept=?, emph: ?, illoc=DEC, mod=?, neg=?, num=?, per=?, recip=?, reflex=?, sta=?, tpl=?, t=?]->ARG[concept=?, macro= A | U | n, num=?, per=?, phrase=?, role: agent | attribute | goal | instrument | location | manner | origin | referent | result | theme, tpl=?, var= x | y | w | z] NUC[concept: ?, emph: ?, illoc:?, mod: ?, num:?, per:?, pol: ? recip:?, reflex:?, sts=?, tpl:?, t: ?] ARG[concept=?, macro= A | U | n, num=?, per=?, phrase=?, role: agent | attribute | goal | instrument | location | manner | origin | referent | result | theme, tpl=?, var= x | y | w | z]

(31c) CORE → ARG NUC ARG ARG (*kernel 3*):
The sensor gives you the measure /It does NOT give ...

The sensor DOES give …
The sensor can/could/must/NOT give…
The sensor cannot give…

STECORE [concept=?, emph: ?, illoc=DEC, mod=?, neg=?, num=?, per=?, recip=?, reflex=?, sta=?, tpl=?, t=?]->ARG[concept=?, macro= A | U | n, num=?, per=?, phrase=?, role: agent | attribute | goal | instrument | location | manner | origin | referent | result | theme, tpl=?, var= x | y | w | z] NUC[concept: ?, emph: ?, illoc:?, mod: ?, num:?, per:?,, pol: ?, recip:?, reflex:?, sta=? tpl:?, t: ?] ARG[concept=?, macro= A | U | n, num=?, per=?, phrase=?, role: agent | attribute | goal | instrument | location | manner | origin | referent | result | theme, tpl=?, var= x | y | w | z] ARG[concept=?, macro= A | U | n, num=?, per=?, phrase=?, role: agent | attribute | goal | instrument | location | manner | origin | referent | result | theme, tpl=?, var= x | y | w | z

To account for imperative sentences in STE, and also in ARTEMIS, we had to create a new syntactic rule. Such rule (see appendix I) is only a modification of the basic rule for the Core minus the initial (subject) argument, which is not present in standard imperative clauses. A further modification is brought by the fact that, according to RRG [Van Valin 2005: 10], the imperative is not marked for tense; therefore, the corresponding attribute on the Nucleus can only be saturated by the value *Null*. The rule for the imperative allows us to parse the three types of kernel sentences both in the positive and in the negative form:

(32a) CORE → NUC:

Do not smoke
Be careful [when you use consumable materials].

CORE [concept=?, emph: ?, illoc=IMP, mod=?, neg=?, num=?, per=?, recip=?, reflex=?, tpl=?, sta=?,t=?]->NUC [concept: ?, emph: ?, illoc:?, mod: ?, neg: ?, num:?, per:?,, recip:?, reflex:?, sta=? tpl:?, t: NULL?]

(32b) CORE → NUC ARG:

Clean the component interface.
Do not breathe the fumes.

CORE [concept=?, emph: ?, illoc=IMP, mod=?, neg=?, num=?, per=?, recip=?, reflex=?, tpl=?, sta=?,t=?]->NUC [concept: ?, emph: ?, illoc:?, mod: ?, neg: ?, num:?, per:?,, recip:?, reflex:?, sta=? tpl:?, t: NULL] ARG[concept=?, macro= A | U | n, num=?, per=?, phrase=?, role: agent | attribute | goal | instrument | location | manner | origin | referent | result | theme, tpl=?, var= x | y | w | z]

(32c) CORE → NUC ARG ARG:

Put the safety barriers in position.
Do not connect the wires to their related pins/sockets.

CORE [concept=?, emph: ?, illoc=IMP, mod=?, neg=?, num=?, per=?, recip=?, reflex=?, tpl=?, sta=?,t=?]->NUC [concept: ?, emph: ?, illoc:?, mod: ?, neg: ?, num:?, per:?,, recip:?, reflex:?, sta=? tpl:?, t: NULL] ARG[concept=?, macro= A | U | n, num=?, per=?, phrase=?, role: agent | attribute | goal | instrument | location | manner | origin | referent | result | theme, tpl=?, var= x | y | w | z] ARG[concept=?, macro= A | U | n, num=?, per=?, phrase=?, role: agent | attribute | goal | instrument | location | manner | origin | referent | result | theme, tpl=?, var= x | y | w | z]

4.3. Construction-L1

Argumental constructions in STE seem to behave exactly the same way as in natural language, therefore, the syntactic rule for the CONSTRL-L1 node in STE is the same Mairal-Usón and Cortés-Rodríguez [2017] devised for ARTEMIS. It has only been slightly adjusted to account for the new attributes of the Core node in STE. As in natural English, L1-constructions in the controlled language can add to the sentence structure either Argument adjuncts (AAJs) or Secondary nuclei (NUC-S):

(33) *Do not keep the bleed valve open* (NUC-S)

CORE [concept=?, emph: ?, illoc=imp, mod=?, neg=?, num=?, per=?, recip=?, reflex=?, tpl=?, sta=?,t=?]
CONSTR-L1[Akt=?, concept=?, illoc=?, mod=?, neg=?, sta: ?, tpl=?, t=?, weight= 0 | 1 | 2 | 3 | 4 | 5 | 6] NUC-S[concept=?, macro= A | U | n, phrase=?, prep=?, role: agent | attribute | goal | instrument | location | manner | origin | refer ent | result | theme, tpl=?, var= x | y | w | z]

(34) *They transmit the vertical loads to the fuselage*(AAJ)

CORE [concept=?, emph: ?, illoc=imp, mod=?, neg=?, num=?, per=?, recip=?, reflex=?, tpl=?, sta=?,t=?] CONSTR-L1[Akt=?, concept=?, illoc=?, mod=?, neg=?, sta: ?, tpl=?, t=?, weight= 0 | 1 | 2 | 3 | 4 | 5 | 6] AAJ[concept=?, macro= A | U | n, phrase=?, prep=?, role: agent | attribute | goal | instrument | location | manner | origin | referent | result | theme, tpl=?, var= x | y | w | z]

4.4. Pre Construction-L1 and Clause

The Pre Construction-L1 (PreC-L1) node in ARTEMIS is a redefinition of the PreCore slot postulated in RRG [Van Valin 2005: 16]. It accounts for fronted elements whose scope includes constructional structures and, in line with RRG, it is realized by question words or by fronted constituents with a pragmatic effect. Since STE does not accept this type of pragmatically marked structures, the PreC-L1 node and its corresponding rule are made redundant. The removal of this node will have an effect on the syntactic rule of the next higher level of the LSC, this is, on STEClause node, whose rule is reformulated as:

(37) STECL[Akt:?, concept=?, emph= ?, Illoc : ?, sta=?, tpl=?, t=?] ->CONSTR-L1 [Akt=?, concept=?, illoc=?, mod=?, tpl=?, t=?, weight= 0 | 1 | 2 | 3 | 4 | 5 | 6] ||

4.5. Right and Left Detached positions. Sentence.

Similar to the PreC-L1 constituent, the LSC in ARTEMIS presents two pragmatically motivated positions in the periphery which are clause external and sentence internal: Left Detached Position (LDP) and Right Detached Position (RDP). Their pragmatic nature implies that in STE the RDP position, which is usually employed as a focus marker, disappears altogether, and the LDP is limited to structures that present initial adjuncts, as in this example:

(40) *After an initial inflation, let the nitrogen pressure become stable*.

Accordingly, the LDP node in STE node needs to be reformulated to restrict its realizational possibilities:

(41) STE LDP [concept=?, macro= A | U | n, num=?, per=?, phrase=?, role: ?, tpl=?, var=]->ADJUNCT [concept=?, phrase=?, prep=?, role: Duration | Frequency | Goal | Instrument | Location | Manner | Means | Origin | Position | Purpose | Quantity | Reason | Result | Scene | Time]

Finally, the disappearance of the RDP node entails rewriting the rule that describes the structure of the sentence as follows:

(42) STE S->CL [Akt:?, concept=?, emph: ?, Illoc : ?, status: ?, tpl=?, t=?] || LDP CL [Akt:?, concept=?, emph: ?, Illoc : ?, status: ?, tpl=?, t=?] ||

Once we have revised the different levels of the LSC, we can now gather the adjustments in this simplified rule for sentence constituency in STE:

(43) STE S-> CL || LDPCL ->CONSTR-L1-> CORE -> NUC-> PRED

STE 100 S ->CL [Akt:?, concept=?, emph= ?, Illoc : ?, sta=?, tpl=?, t=?] STE LDP [concept=?, macro= A | U | n, num=?, per=?, phrase=?, role: ?, tpl=?, var=]-> CL [Akt:?, concept=?, emph= ?, Illoc : ?, sta=?, tpl=?, t=?]||STE LDP [concept=?, macro= A | U | n, num=?, per=?, phrase=?, role: ?, tpl=?, var=]->ADJUNCT [concept=?, phrase=?, prep=?, role: Duration | Frequency | Goal | Instrument | Location | Manner | Means | Origin | Position | Purpose

| Quantity | Reason | Result | Scene | Time] CL [Akt:?, concept=?, emph= ?, Illoc : ?, sta=?, tpl=?, t=?]->CONSTR-L1 [Akt=?, concept=?, illoc=?, mod=?, tpl=?, t=?, weight= 0 | 1 | 2 | 3 | 4 | 5 | 6] ||STECORE [illoc=DEC, mod=?, neg=?, num=?, per=?, recip=?, reflex=?, sta=?, tpl=?, t=?]->ARG NUC || ARG NUC ARG ||ARG NUC ARG ARG||STECORE [illoc=IMP, mod=?, neg=?, num=?, per=?, recip=?, reflex=?, sta=?, tpl=?, t=?] ->NUC || NUC ARG ||NUC ARG ARG->SD- STE 100 NUC [concept: ?, emph: ?, illoc:?, mod: ?, neg: ?, num:?, per:?, recip:?, reflex:?, sta=? tpl:?, t: ?] → PRED[concept= ?, illoc=?, num=?, per=?, sta= ? tpl=?, t=?] ||

Conclusion

In this paper we have discussed the adjustments necessary for the parsing of simple sentences in the controlled technical natural language STE within ARTEMIS. These rules had to be accommodated to the restrictions imposed by the communicative functions associated with instructions manuals. The discursive nature of this type of texts and the standardization imposed by the STE specification manual have syntactic and pragmatic effects that need to be accounted for in ARTEMIS, as we have tried to illustrate with the analysis presented in this study. The implementation of a linguistically motivated deep parser such as ARTEMIS is not a straightforward task and further research in areas such as the treatment of hypotaxis and parataxis, the integration of adjuncts or negation – which are complex enough to deserve a separate analysis –is needed to achieve a complete description of sentential structures in STE.

References

ASD-STE Simplified Technical English Specification STE-100.International specification for the preparation of technical documentation in a controlled language. Issue 7, January 2017

Cortés-Rodríguez, F. (2016). Towards the computational implementation of Role and Reference Grammar: Rules for the Syntactic Parsing of RRG phrasal constituents. Círculo*,* 65, 75-108.

Cortés-Rodríguez, F. and Mairal-Usón, R. (2016). Building an RRG computational Grammar. Onomazein 34, 86-117

Díaz-Galán, A. & Fumero-Pérez, M. (2016).Developing Parsing Rules within ARTEMIS: The case of Do Auxiliary Insertion. In C. Periñán Pascual& E. Mestre Mestre (Eds.*),* Understanding Meaning and Knowledge Representation: From Theoretical and Cognitive linguistics to Natural Language processing. (pp. 283-302). Cambridge: Cambridge Scholars Publishing.

Díaz-Galán, A. & Fumero-Pérez, M. (2017). ARTEMIS: State of the Art and Future Horizons. Revista de Lenguas para Fines Específicos 23/1, 16-40.

Fumero-Pérez, M. & Díaz-Galán, A. (2017). The Interaction of Parsing Rules and Argument-Predicate constructions: Implications for the structure of the Grammaticon in FunGramKB. Revista de Lingüística y Lenguas Aplicadas 12: https://doi.org/10.4995/rlyla.2017.5406

Kittredge, R. (2003). Sublanguages and controlled languages. In M. Ruslan (ed.), The Oxford Handbook of Computational Linguistics (pp. 430-447). Oxford: O.U.P.

Khun, Tobias (2014). A survey and classification of Controlled Natural Languages. Computational Linguistics, volume 40, 1.

Mairal-Usón, R. and Cortés-Rodríguez, F. (2017). Automatically Representing Text Meaning via an Interlingua-based System (ARTEMIS). A further step towards the computational representation of RRG. *Journal of Computer-Assisted Linguistic Research*1, 61-87.

Martín-Díaz, M.A. (2017). An account of English yes/no interrogative sentences within ARTEMIS.Revista de Lenguas para Fines Específicos 23/1, 41-62.

Mayes, P. (2003). A genre analysis of cooking clases in Japan and America. John Benjamins.

Nivre, J. (2010). Handbook of natural language processing second edition. Boca Ratón: Taylor and Francis Group.

Periñán Pascual, C. and Arcas Túnez, F. (2007). Cognitive modules of an NLP knowledge base for language understanding". Procesamiento del Lenguaje Natural 39, 197-204.

Periñán-Pascual, C. & Arcas Túnez, F. (2014). The implementation of the CLS constructor in ARTEMIS. In B. Nolan & C. Periñán-Pascual, (Eds.), Language Processing and Grammars the role of functionally oriented computational models (pp. 164-196) Amsterdam / Philadelphia: John Benjamins.

Periñán-Pascual, C. &Mairal-Usón, R. (2010). Bringing Role and Reference Grammar to natural language understanding. *Procesamiento del Lenguaje Natural*, 43, 265-273.

Quirk, R., Greenbaum S., Leech, G. & Svartvik, J. (1985). A comprehensive grammar of the English language. London: Longman.

Ruiz de Mendoza Ibáñez, F.&Mairal-Usón, R. (2008). Levels of description and constraining factors in meaning construction: an introduction to the Lexical Constructional Model. Folia Linguistica, 42(2), 355-400.

Sag, I., Wasow, T. & Bender, E. (2003). Syntactic Theory: Formal Introduction. Stanford: CSLI Publications

Sharpe, M. (2014). Language forms and rhetorical functions in technical instructions English for Specific Purposes World, Issue 43, Vol. 15.

Schwitter, R. (2010). Controlled Natural Languages for Knowledge Representation. In: Proceedings of COLING 2010, (pp. 1113-1121). Beijing, China.

Van Valin, R. (2005). Exploring the Syntax-Semantics Interface. Cambridge: Cambridge University Press.

Van Valin, R. & La Polla, R. (1997). Syntax. Cambridge: Cambridge University Press.

APPENDIX I:
RULES FOR THE LSC IN STE

1. Rule for PRED (Cortés-Rodríguez and Mairal-Usón, 2016:107):

PRED[concept= ?, illoc=?, num=?, per=?, tpl=?, t=?] ->VERB[concept=?, recip=?, reflex=?, t=?]

2. Syntactic rule for STE NUCLEUS (Adapted from Cortés-Rodríguez and Mairal-Usón, 2016:107):

AUX [emph= e ∣ null, num= pl ∣ sg ∣ null, per= 1 ∣ 2 ∣ 3, syn= verb ∣ null, t= pres ∣ past ∣ fut ∣ null] PRED [concept= ?, illoc=?, num=?, per=?, tpl=?, t=?] ||AUX [emph= e ∣ null, num= pl ∣ sg ∣ null, per= 1 ∣ 2 ∣ 3, syn= verb ∣ null, t= pres ∣ past ∣ fut ∣ NEG[assr: neg]PRED [concept= ?, illoc=?, num=?, per=?, tpl=?, t=?] ||MODD [mod= abl ∣ obl ∣ perm, pol: pos ∣ neg, syn= verb ∣ null, t= pres ∣ past ∣ fut ∣ null] PRED [concept= ?, illoc=?, num=?, per=?, tpl=?, t=?] ||MODD [mod= abl ∣ obl ∣ perm, pol: pos ∣ neg, syn= verb ∣ null, t= pres ∣ past ∣ fut ∣ null] NEG [assr: neg] PRED[concept= ?, illoc=?, num=?, per=?, tpl=?, t=?]MODST [mod= poss, syn= verb ∣ null, t= pres ∣ past ∣ fut ∣ null] PRED[concept= ?, illoc=?, num=?, per=?, tpl=?, t=?]

3. Syntactic rule for STE CORE in DECLARATIVES (Adapted from Mairal-Usón and Cortés Rodríguez 2017)

STE CORE [concept=?, emph: ?, illoc=DEC, mod=?, neg=?, num=?, per=?, recip=?, reflex=?, sta=?, tpl=?, t=?]->ARG[concept=?,macro= a ∣ u ∣ n, num=?, per=?, phrase=?, role: agent ∣ attribute ∣ goal ∣ instrument ∣ location ∣ manner ∣ origin ∣ referent ∣ result ∣ theme, tpl=?, var= x ∣ y ∣ w ∣ z] NUC [concept: ?, emph: ?, illoc:?, mod: ?, neg: ?, num:?, per:?,recip:?, reflex:?, tpl:?, sta=?t: ?] || ARG[concept=?, macro= A ∣ U ∣ n, num=?, per=?, phrase=?, role: agent ∣ attribute ∣ goal ∣ instrument ∣ location ∣ manner ∣ origin ∣ referent ∣ result ∣ theme, tpl=?, var= x ∣ y ∣ w ∣ z] NUC[concept: ?, emph: ?, illoc:?, mod: ?, num:?, per:?, pol: ? recip:?, reflex:?, sts=?, tpl:?, t: ?] ARG[concept=?, macro= A ∣ U ∣ n, num=?, per=?, phrase=?, role: agent ∣ attribute ∣ goal ∣ instrument ∣ location ∣ manner ∣ origin ∣ referent ∣ result ∣ theme, tpl=?, var= x ∣ y ∣ w ∣ z] || ARG[concept=?, macro= A ∣ U ∣ n, num=?, per=?, phrase=?, role: agent ∣ attribute ∣ goal ∣ instrument ∣ location ∣ manner ∣ origin ∣ referent ∣ result ∣ theme, tpl=?, var= x ∣ y ∣ w ∣ z] NUC[concept: ?, emph: ?, illoc:?, mod: ?, num:?, per:?,, pol: ?, recip:?, reflex:?, sta=? tpl:?, t: ?] ARG[concept=?, macro= A ∣ U ∣ n, num=?, per=?, phrase=?, role: agent ∣ attribute ∣ goal ∣ instrument ∣ location ∣ manner ∣ origin ∣ referent ∣ result ∣ theme, tpl=?, var= x ∣ y ∣ w ∣ z] ARG[concept=?, macro= A ∣ U ∣ n, num=?, per=?, phrase=?, role: agent ∣ attribute ∣ goal ∣ instrument ∣ location ∣ manner ∣ origin ∣ referent ∣ result ∣ theme, tpl=?, var= x ∣ y ∣ w ∣ z

4. Syntactic rule for the STE-100 CORE in IMPERATIVES:

STE CORE [concept=?, emph: ?, illoc=IMP, mod=?, neg=?, num=?, per=?, recip=?, reflex=?, tpl=?, sta=?,t=?]->NUC [concept: ?, emph: ?, illoc:?, mod: ?, neg: ?, num:?, per:?,, recip:?, reflex:?, sta=? tpl:?, t: NULL] ||NUC [concept: ?, emph: ?, illoc:?, mod: ?, neg: ?, num:?, per:?,, recip:?, reflex:?, sta=? tpl:?, t: NULL] ARG[concept=?, macro= A ∣ U ∣ n, num=?, per=?, phrase=?, role: agent ∣ attribute ∣ goal ∣ instrument ∣ location ∣ manner ∣ origin ∣ referent ∣ result ∣ theme, tpl=?, var= x ∣ y ∣ w ∣ z] || NUC [concept: ?, emph: ?, illoc:?, mod: ?, neg: ?, num:?, per:?,, recip:?, reflex:?, sta=? tpl:?, t: NULL] ARG[concept=?, macro= A ∣ U ∣ n, num=?, per=?, phrase=?, role: agent ∣ attribute ∣ goal ∣ instrument ∣ location | manner | origin | referent | result | theme, tpl=?, var= x | y | w | z] ARG[concept=?, macro= A | U | n, num=?, per=?, phrase=?, role: agent | attribute | goal | instrument | location | manner | origin | referent | result | theme, tpl=?, var= x | y | w | z]

5. Syntactic rule for STE-100L1-CONSTRUCTION (Adapted from Mairal-Usón and Cortés-Rodríguez, 2017):

CONSTR-L1[Akt=?, concept=?, illoc=?, mod=?, neg=?, sta: ?, tpl=?, t=?, weight= 0 | 1 | 2 | 3 | 4 | 5 | 6] ->STE-100 CORE [concept=?, emph: ?, illoc=imp, mod=?, neg=?, num=?, per=?, recip=?, reflex=?, tpl=?, sta=?,t=?]|| STE-100 CORE [concept=?, emph: ?, illoc=imp, mod=?, neg=?, num=?, per=?, recip=?, reflex=?, tpl=?, sta=?,t=?]AAJ[concept=?, macro= A | U | n, phrase=?, prep=?, role: agent | attribute | goal | instrument | location | manner | origin | referent | result | theme, tpl=?, var= x | y | w | z] || ->STE-100 CORE [concept=?, emph: ?, illoc=imp, mod=?, neg=?, num=?, per=?, recip=?, reflex=?, tpl=?, sta=?,t=?] NUC-S [concept=?, macro= A | U | n, phrase=?, prep=?, role: agent | attribute | goal | instrument | location | manner | origin | referent | result | theme, tpl=?, var= x | y | w | z] || CONSTR-L1[Akt=?, concept=?, illoc=?, mod=?, neg=?, sta: ?, tpl=?, t=?, weight= 0 | 1 | 2 | 3 | 4 | 5 | 6] AAJ[concept=?, macro= A | U | n, phrase=?, prep=?, role: agent | attribute | goal | instrument | location | manner | origin | referent | result | theme, tpl=?, var= x | y | w | z] || CONSTR-L1[Akt=?, concept=?, illoc=?, mod=?, neg=?, sta: ?, tpl=?, t=?, weight= 0 | 1 | 2 | 3 | 4 | 5 | 6] NUC-S[concept=?, macro= A | U | n, phrase=?, prep=?, role: agent | attribute | goal | instrument | location | manner | origin | referent | result | theme, tpl=?, var= x | y | w | z] || CONSTR-L1[Akt=?, concept=?, illoc=?, mod=?, neg=?, tpl=?, t=?, weight= 0 | 1 | 2 | 3 | 4 | 5 | 6] AAJ[concept=?, macro= A | U | n, phrase=?, prep=?, role= agent | attribute | goal | instrument | location | manner | origin | referent | result | theme, tpl=?, var= w] NUC-S [concept=?, macro= A | U | n, phrase=?, prep=?, role= agent | attribute | goal | instrument | location | manner | origin | referent | result | theme, tpl=?, var= v]

6. Syntactic rule for STE-100 CLAUSE:

STECL [Akt:?, concept=?, emph= ?, Illoc : ?, sta=?, tpl=?, t=?] ->CONSTR-L1 [Akt=?, concept=?, illoc=?, mod=?, tpl=?, t=?, weight= 0 | 1 | 2 | 3 | 4 | 5 | 6] ||

7. Syntactic rule for STE-100 LEFT DETACHED POSITION (LDP):

STE LDP [concept=?, macro= A | U | n, num=?, per=?, phrase=?, role: ?, tpl=?, var=]->ADJUNCT [concept=?, phrase=?, prep=?, role: Duration | Frequency | Goal | Instrument | Location | Manner | Means | Origin | Position | Purpose | Quantity | Reason | Result | Scene | Time]

8. Syntactic rule for STE-100 SENTENCE:

STE S—>CL [Akt:?, concept=?, emph: ?, Illoc : ?, status: ?, tpl=?, t=?] || LDP CL [Akt:?, concept=?, emph: ?, Illoc : ?, status: ?, tpl=?, t=?] ||

ПОДРОБНЫЙ СИНТАКСИЧЕСКИЙ АНАЛИЗ ЯЗЫКА АВИАЦИОННОЙ ПРОМЫШЛЕННОСТИ: НАСТРОЙКА АНАЛИЗАТОРА ARTEMIS ДЛЯ ОБРАБОТКИ ПРОСТЫХ ПРЕДЛОЖЕНИЙ ЯЗЫКА ASDSTE-100

А. Диас-Галан
Университет Ла Лагуны, Институт лингвистики имени Андреса Бельо (Тенерифе, Испания)
adiazgal@ull.edu.es

Лингвистический анализатор ARTEMIS (Automatically Representing Text Meaning via an Interlingua-based System) был разработан с целью синтаксического, семантического и прагматического анализа фрагментов естественных языков [Периньян-Паскуаль и Аркас Туньес 2014]. Обычной практикой при тестировании анализатора перед тем, как начать использовать его для анализа естественных языков является его применение к контролируемому языку. В данной статье мы исследуем анализ простых предложений языка ASD-STE 100 (упрощенный техниче-

ский английский Европейской ассоциации предприятий аэрокосмической и оборонной промышленности). Мы создадим или пересмотрим лексические и синтаксические правила для анализатора ARTEMIS, чтобы настроить анализатор с учетом формальных требований и ограничений, а также ограниченных коммуникативных функций, присущих этому языку. Эти правила соответствуют принципам двух лингвистических моделей, на которые опирается анализатор ARTEMIS – Референциально-Ролевой грамматики и Лексико-Конструктивисткой модели. С помощью этих правил можно будет определить и описать в системе ARTEMIS каждый из узлов, из которых состоят простые предложения, и таким образом приспособить анализатор к языку ASD-STE 100. В заключение мы отметим области, нуждающиеся в дополнительных исследованиях для полноценного использования анализатора.

Ключевые слова: *база знаний по функциональной лингвистике (FunGramKB), ARTEMIS, обработка естественного языка (NLP), упрощенный технический английский Европейской ассоциации предприятий аэрокосмической и оборонной промышленности.*

Для цитирования: *Díaz-Galán A.* Deep Parsing for the aviation industry: Adjusting ARTEMIS for parsing simple clauses in ASD STE-100 // Вопросы когнитивной лингвистики. 2018. № 3. С. 83-96.