JOURNAL OF ENGLISH STUDIES – VOLUME 17 (2019), 149-174. http://doi.org/10.18172/jes.3935

DESIGNING THE LEXICAL RULES FOR THE PARSING OF ASD-STE100 FUNCTION WORDS IN ARTEMIS FROM A ROLE AND REFERENCE GRAMMAR PERSPECTIVE[1]

María del Carmen Fumero-Pérez
Instituto de Lingüística Andrés Bello, Universidad de La Laguna
mfumero@ull.edu.es

Ana Díaz-Galán
Instituto de Lingüística Andrés Bello, Universidad de La Laguna
adiazgal@ull.edu.es

ABSTRACT. *ARTEMIS (Automatically Representing Text Meaning via an Interlingua-based System), is a natural language processing device, whose ultimate aim is to be able to understand natural language fragments and arrive at their syntactic and semantic representation. Linguistically, this parser is founded on two solid linguistic theories: the Lexical Constructional Model and Role and Reference Grammar. Although the rich semantic representations and the multilingual character of Role and Reference Grammar make it suitable for natural language understanding tasks, some changes to the model have proved necessary in order to adapt it to the functioning of the ARTEMIS parser. This paper will deal with one of the major modifications that Role and Reference Grammar had to undergo in this process of adaptation, namely, the substitution of the operator projection for feature-based structures, and how this will influence the description of function words in ARTEMIS, since they are strongly responsible for the encoding of the grammatical information which in Role and Reference Grammar is included in the operators. Currently, ARTEMIS is being implemented for the controlled natural language ASD-STE100, the Aerospace and Defence Industries Association of Europe Simplified Technical English, which is an international specification for the preparation of technical documentation in a controlled language. This controlled language is used in the belief that its simplified nature makes it a good corpus to carry out a preliminary testing of the adequacy of the parser. In this line, the aim of this work is to create a catalogue of function words*

[1] This work has been developed within the framework of the research project "Desarrollo de un laboratorio virtual para el procesamiento computacional de la lengua desde un paradigma functional" (UNED) FF2014-53788-C3-1-P, funded by the Spanish Ministry of Economy and Competitiveness.

in ARTEMIS for ASD-STE100, and to design the lexical rules necessary to parse the simple sentence and the referential phrase in this controlled language.

Keywords: Role and Reference Grammar (RRG), ARTEMIS, parser, ASD-STE100, controlled natural language, natural language processing (NLP), function words, lexical rules.

DISEÑO DE REGLAS LÉXICAS PARA EL PARSEADO DE LAS PALABRAS FUNCIONALES DE ASD-STE100 EN ARTEMIS DESDE LA PERSPECTIVA DE LA GRAMÁTICA DEL PAPEL Y LA REFERENCIA

RESUMEN. *ARTEMIS (Automatically Representing Text Meaning via an Interlingua-based System) es un dispositivo para el procesamiento de lenguaje natural cuyo propósito es procesar fragmentos de lenguaje natural y llegar a producir su representación sintáctica y semántica. Desde un punto de vista lingüístico, este parseador se fundamenta en dos sólidos modelos lingüísticos: el Modelo Léxico Construccional y la Gramática del Papel y la Referencia. Aunque las ricas representaciones semánticas y el carácter multilingüe que caracterizan a la Gramática del Papel y la Referencia la hacen adecuada para tareas relacionadas con el procesamiento de lenguaje natural, ha sido necesario llevar a cabo algunos cambios en el modelo para adaptarlo al funcionamiento del parseador ARTEMIS. Este trabajo se centrará en uno de los cambios más relevantes en este proceso de adaptación de la Gramática del Papel y la Referencia: la sustitución de la proyección de operadores por estructuras de rasgos y la influencia que esto tiene en la descripción de las palabras funcionales en ARTEMIS, al ser estas palabras las encargadas de codificar en gran medida la información gramatical que en la Gramática del Papel y la Referencia se incluye en los operadores. En este momento, ARTEMIS está siendo implementado para el lenguaje controlado ASD-STE100, el inglés técnico simplificado empleado para la preparación de documentación técnica por la Asociación de la Industria Aeroespacial y de Defensa Europeas. Este lenguaje es usado bajo la asunción de que su naturaleza simplificada lo convierte en un buen corpus para probar la adecuación del parseador. En esta línea, el objetivo de este trabajo es crear un catálogo de palabras funcionales en ARTEMIS para ASD-STE100 y diseñar las reglas léxicas necesarias para el parseado de la oración simple y el sintagma referencial en este lenguaje controlado.*

Palabras clave: Gramática del Papel y la Referencia (GPR), parseador, ARTEMIS, lenguaje natural controlado, ASD_STE100, procesamiento de lenguaje natural (PLN), palabras funcionales, reglas léxicas.

Received 18 December 2018
Revised version accepted 5 June2019

1. INTRODUCTION

The Automatically Representing Text Meaning via an Interlingua-based System (ARTEMIS) is a parser designed by Periñán-Pascual and Arcas-Túnez (2010, 2014);

Periñán-Pascual and Mairal-Usón (2010a, 2010b) for Natural Language Processing (NLP) tasks, whose ultimate aim is to process natural language fragments and produce their corresponding syntactic and semantic representation. With this aim, the prototype draws from two functional linguistic models: the Lexical Constructional Model (Ruiz de Mendoza and Mairal-Usón 2008; Mairal Usón and Ruiz de Mendoza 2009) and Role and Reference Grammar (Van Valin and LaPolla 1997; Van Valin 2005). The linking algorithm (form-to-meaning and meaning-to-form), the rich semantic representations and its multilingual character are put forward by Van Valin to defend the suitability of the Role and Reference Grammar (RRG) linguistic theory for Natural Language Understanding (NLU) tasks. In the literature, we may find several authors who deal with the computational implementation of RRG. These include Osswald and Kallmeyer (forthcoming), who propose a formalization of RRG to make it computationally adequate, or Ball (2017), who defends a NLU model based on RRG and the brain based Patom theory, a cognitive model conceived to discover how a brain functions so that it can be imitated by a machine. The actual development of a parser based on RRG was the aim of the research carried out by Diedrichsen (2014), who designed a sentence parser for German, or by Guest (2008), who showed, through the analysis of sentences from student work, how RRG can be implemented into a standard rule-based parser. Within the specific framework of ARTEMIS, a relevant number of works have also been published dealing with the adjustments that have revealed necessary to adapt the RRG model to the functioning of the parser (Cortés-Rodríguez 2016; Cortés-Rodríguez and Mairal-Usón 2016; Martín-Díaz 2017; Díaz-Galán and Fumero-Pérez 2016; Fumero-Pérez and Díaz-Galán 2017; Rodríguez-Juárez 2017). In this paper we will address one of the substantial modifications which the RRG model had to undergo in this adaptation process, namely, the substitution of the operator projection for feature-based structures, and how this influences the description of function words in ARTEMIS. These functional items, which are heavily responsible for the encoding of grammatical information of the kind represented originally by operators in RRG, play a vital role in the parsing process. Incorporating such words in ARTEMIS involves the creation of a fully-fledged catalogue of Parts of Speech (POS), together with a formal description of the function words that instantiate them. Following the paradigm of Unification-based or Feature-based grammars (Sag, Wasow and Bender 2003), this is, grammars which describe language structures by means of formal features which are later unified, in ARTEMIS, the description of grammatical categories such as tense, aspect, modality or illocutionary force —which are part of the operator projection in RRG— has to be done through grammar formalisms which list their features and values, the so-called Attribute Value Matrices (AVMs). Additionally, to provide the parser with a computer-interpretable characterization of function words, lexical rules have to be designed, since they are the means of encoding the relevant morphosyntactic information attached to each functional item which will later be integrated into the higher syntactic structures where they participate. Within ARTEMIS, as we will see in section 2, these formal descriptions are stored in the Grammar Development Environment (GDE).

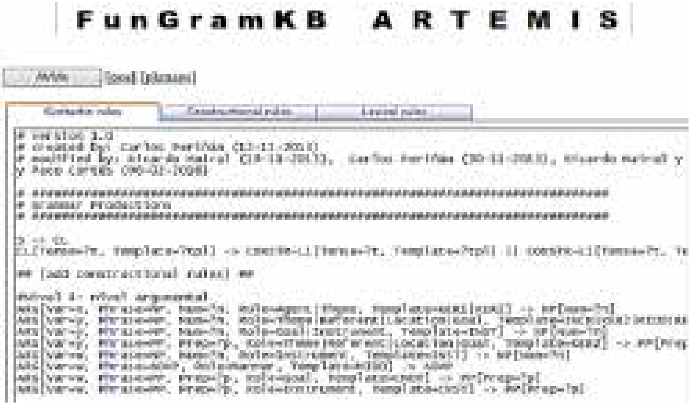
At this stage, ARTEMIS is being implemented for the controlled natural language ASD-STE100, the Aerospace and Defence Industries Association of Europe Simplified Technical English. Given its simplified nature, this controlled language is being used as a test bench to achieve the eventual parsing of natural English. ASD-STE100 is described in the International Specification for the Preparation of Technical Documentation in a Controlled Language (issue 7, January 2017). The specification provides writing guidelines to be followed by technical writers and it also includes a controlled dictionary of "approved" vocabulary, together with its "approved" forms. This writing manual, therefore, imposes constraints on morphology, syntax and vocabulary which will restrict the members of the Catalogue of POS in ARTEMIS in its version for ASD-STE100. Thus, it is the purpose of this study to redesign the catalogue of function words and to provide the AVMs and lexical rules necessary to parse both the simple clause and the referential phrase in this simplified language within the framework of RRG.

In the following section we will briefly outline the performance of the ARTEMIS parser and the role of function words in the GDE in ARTEMIS. In section 3 we will focus on the changes to the RRG model required for an effective computational treatment of such words in ASD-STE100. In sections 4 and 5 we will present a catalogue of function words consistent with the specific characteristics of both the simple clause and the referential phrase in ASD-STE100. Finally, conclusions will be presented in section 6.

2. THE ARTEMIS PARSER

Three are the main elements that structure ARTEMIS: the Grammar Development Environment (GDE); the Conceptual Logical Structure Constructor and the COREL-Scheme Builder. The GDE comprises two types of constructs:

A catalogue of the AVMs which describe grammatical units, and a set of production rules (syntactic, lexical and constructional) necessary to generate a feature-based grammar. It is, therefore, both a repository of AVMs and a computational grammar which allows the morphosyntactic representation of natural language fragments. The following figure shows a screen capture of the GDE, as it appears in ARTEMIS:



```
# version 1.0
# created by: Carlos Periñán [illegible]
# modified by: Ricardo Mairal [illegible], Carlos Periñán [illegible], Ricardo Mairal y
y Paco Cortés [illegible]

# ################################################################
# Grammar Productions
# ################################################################

S -> CL
CL[Tense=?t, Template=?tpl] -> CORE-CL[Tense=?t, Template=?tpl] || CORE-CL[Tense=?t, Te

## [add constructional rules] ##

#Nivel 1- nivel argumental
ARG[Var=x, Phrase=NP, Num=?n, [illegible]] -> NP[Num=?n]
ARG[Var=y, Phrase=NP, Num=?n, [illegible]
ARG[Var=y, Phrase=NP, Num=?n, [illegible]] -> NP[Num=?n]
ARG[Var=y, Phrase=PP, Prep=?p, [illegible]] -> PP[Prep
ARG[Var=w, Phrase=NP, Num=?n, [illegible]] -> NP[Num=?n]
ARG[Var=w, Phrase=ADVP, [illegible]] -> ADVP
ARG[Var=w, Phrase=PP, Prep=?p, [illegible]] -> PP[Prep=?p]
ARG[Var=w, Phrase=PP, Prep=?p, Role=Instrument, [illegible]] -> PP[Prep=?p]
```

Figure 1. The ARTEMIS Grammar Development Environment (GDE).

The second component —the Conceptual Logical Structure Constructor— is in charge of generating the underlying semantic structure of the input text. The resulting representation, the Conceptual Logical Structure or CLS (see example 2), is basically a computational implementation of RRG's Logical Structures. Finally, since ARTEMIS is one of the applications within the FunGramKB[2] suite, the COREL-Scheme Builder has to transduce the CLS into COREL, the formal language of the FunGramKB environment. The parsing routine in ARTEMIS has been designed for these three components to interact as represented in the UML diagram illustrated in figure 2 (Periñán-Pascual and Arcas-Túnez 2014: 178).

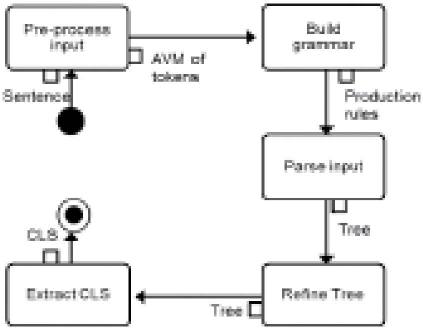
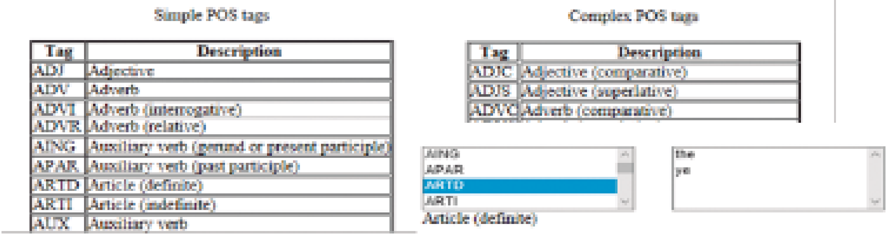


Figure 2. UML diagram of the parsing routine in ARTEMIS.

[2] For a full description of FunGramKB the reader is referred to Periñán-Pascual and Arcas-Túnez (2010), and Periñán-Pascual (2012).

The diagram shows how the parsing starts with the extraction of word tokens, a pre-processing phase that separates the input text into basic units of analysis, which are then labelled with a part of speech tag described in the form of Attribute Value Matrices (AVMs). This part of speech tagging provides the first morphosyntactic characterization of the lexical items. Content words, which are associated with a concept from the FunGramKB Core Ontology or Satellite Ontology (in the case of technical words), are listed and described in an English Lexicon. Function words, on the other hand, are merely assigned a POS tag from lemmatization NLP libraries, a procedure which, in our opinion, is linguistically insufficient. Figure 3 provides an example of the scarce information concerning POS available in ARTEMIS to date.



Simple POS tags

Tag	Description
ADJ	Adjective
ADV	Adverb
ADVI	Adverb (interrogative)
ADVR	Adverb (relative)
AING	Auxiliary verb (gerund or present participle)
APAR	Auxiliary verb (past participle)
ARTD	Article (definite)
ARTI	Article (indefinite)
AUX	Auxiliary verb

Complex POS tags

Tag	Description
ADJC	Adjective (comparative)
ADJS	Adjective (superlative)
ADVC	Adverb (comparative)

AING
APAR
ARTD
ARTI

Article (definite)

the
ye

Figure 3. POS in ARTEMIS.

The Build Grammar stage involves the application of the three types of rules stored in the GDE component. The first, syntactic rules, are designed by the linguist and describe the internal constituency of each of the nodes of the constituent projection of the clause and the phrase. As an example, (1) shows the simplified rule for the possible categories that can occupy the initial node in the noun phrase (referential phrase in RRG terms). The rule shows how this referential phrase initial position (RPIP) may be instantiated by articles (ART), demonstratives (DETD), numerals (DETNC or DETNO), quantifiers (DETQ) and their various combinations. The options also include the possibility of having phrases in RPIP.

(1) RPIP → ART DETD DETNC || DETNO DETP DETQ DETQ ART DETQ DETD DETQ DETP MP RP

The second type of production rules, constructional rules, governs the integration of argumental constructions —those which are based on predicate argument relationships (Goldberg 1995)— in the Layered Structure of the Clause (LSC) by describing their morphological and semantic characteristics. As we will see in detail later, lexical rules spell out the morphosyntactic information relevant to each of the word tokens. Periñán-Pascual and Arcas-Túnez affirm that "unlike syntactic rules, which users can pre-define through the Grammar Development Environment, constructional and lexical rules are created dynamically at runtime". That is, "ARTEMIS will build only those constructional and lexical rules which can be directly derived from the constructional schemata and lexical entries being linked to the predicates in the input stream" (2014: 181). In the case of constructional meaning, it is true that ARTEMIS can resort both to the FunGramKB Lexicon to check for the constructional combinations of a given predicate and to the FunGramKB Grammaticon to find the corresponding constructional schema. However, with respect to function words, the information stored in the GDE, as it is only sourced from lemmatization libraries, is not rich enough as to allow the parser to create the lexical rules at runtime. This deficiency calls for the creation of a whole catalogue of function words, together with their lexical rules. Such rules correspond to the specific realization of a given grammatical category, whose attributes have to be described in the form of AVMs. The catalogue of POS related to function words in ARTEMIS must, therefore, be revised and systematized in such a way that they can be effectively inserted into the syntactic rules and provide the relevant information previously conveyed by some of the operators in RRG.

The third phase in the parsing routine is the creation of a parse tree in which the parser will carry out a feature unification process in a bottom-up fashion, which may affect the whole structure of the clause in such a way that, as shown in figure 4, the scope of an operator as is illocutionary force may start in the first terminal constituent of the nucleus (auxiliary verb *has*) and percolate up to the sentence node.

Finally, once the parser has yielded the parse tree, as a final step, the CLS constructor will generate a semantically enriched conceptual representation of the input sentence, as illustrated in example (2) for the sample sentence *Louise had baked a cake for the kids* (Fumero-Pérez and Díaz-Galán 2017: 38). The inclusion of the higher level clausal operators tense and illocutionary force in the final formal representation is proof of the relevance of operators for the parser[3].

[3] Fumero Pérez and Díaz Galán (2017: 38) illustrate the whole parsing process:

Sample sentence:	Louise had baked a cake for the kids.
RRG Logical Structure:	[[do´(Louise, Ø)] CAUSE [BECOME baked´(cake)]] PURP [BECAME have´(the kids, cake)]
CLS:	<ifdecl<tensepast<constr_l1fben<constr_l1ker2<aktcacc<[+BAKE_00(%LOUISE_00-Agent,+CAKE_00-Referent, +CHILD_00-Beneficiary)] >>>>>>
COREL scheme:	+(e1: +BAKE_00 (x1: %LOUISE_00)theme (x2: +CAKE_00)referent (f1: (e2: +DO_00 (x1)Agent (e1)Referent (f2: +CHILD_00)Beneficiary))Purpose)

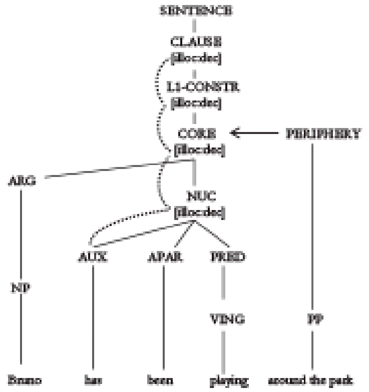


Figure 4. Feature unification path of illocutionary force in the LSC. (Mairal-Usón and Cortés-Rodríguez 2017: 66).

(2) CLS:<IFDECL<TensePAST<CONSTR_L1FBEN<CONSTR_L1KER2<AKTCACC <[+BAKE_00(%LOUISE_00-Agent, +CAKE_00-Referent, +CHILD_00-Beneficiary)] >>>>>[4]

3. THE OPERATOR PROJECTION IN ARTEMIS

As stated above, several efforts have been made to adapt the description of clausal and phrasal constituents in RRG to the functioning of the ARTEMIS parser. At clause level, some of the most relevant changes have been suggested by Periñán-Pascual (2013), who initially proposes a modified or enhanced version of the LSC in RRG which adds a new L1-CONSTRUCTION node to account for argumental constructions. Mairal-Usón and Cortés-Rodríguez (2017) further refine the LSC by adding a Pre-Construction L1 node and also by designing the syntactic rules which describe each of the nodes of this modified Layered Structure of the Clause in ARTEMIS. At phrase level, Cortés-Rodríguez (2016) offers a description of the layered structure of referential and modifier phrases as proposed in Van Valin (2008), which in turn replaces Van Valin and LaPolla's (1997) and Van Valin's (2005) Layered Structure of Noun and adjective phrases. Later Cortés-Rodríguez (2016) designs the set of syntactic rules necessary for parsing such phrasal structures.

[4] In natural language terms, this CLS represents the event as a declarative (IFDECL) in the past (Tense PAST) which presents a monotransitive verb (CONSTR_L1KER2), and a *for benefactive* construction (FBEN). The aspectual value of the event corresponds to a causative accomplishment (AKTCACC). Notice also how the CLS substitutes the lexical items with the corresponding FunGramKB ontological concepts (bake =+BAKE_00, etc.) and assigns them a thematic role.

He offers a reinterpretation of the operator projection as feature bearing devices which include morphosyntactic information in the form of AVMs, both for the clause and the phrase. At clause level, this formalization applies to each of the operators described by Van Valin (2005: 12) for the different nodes of the LSC in English. These operators may affect different parts of the clause, in such a way that the operator aspect (perfect, progressive) has scope over the Nucleus (NUC) node; the modality (deontic) and negation operators modify the Core node; and, finally, status (epistemic modality), tense, and illocutionary force operators have the Clause in their scope. Example (3) illustrates how the compulsory operator illocutionary force, can be formalized as an AVM which presents three mutually exclusive possible values: declarative, imperative or interrogative.

```
(3) <Attribute ID= “Illocutionary Force” obl= “+” num= “1”>
        <Value Tag=”declarative”>dec</Value>
        <Value Tag=”imperative”>imp</Value>
        <Value Tag=”interrogative”>int</Value>
    </Attribute>
```

The same occurs in the operator projection of the referential phrase (RP), where, again, each of the layers can be modified by a number of operators in English. The NuclearRP node is only affected by nominal aspect (count-mass distinction), while the operators that influence the CoreRP are number, quantification (quantifiers) and negation. Finally, the RP level operators include definiteness and deixis. Example (4) shows how the RP operator definiteness is formalized in an AVM.

```
(4)  <Attribute ID= “Def” obl=”+” num=”1”>
        <Value>?def</Value>
        <Value Tag=”definite”>d</Value>
        <Value Tag=”indefinite”>i</Value>
     </Attribute>
```

Similar to the role of illocutionary force at clause level, RP operators such as definiteness modify the whole RP and ground it in the discourse (Van Valin, 2005: 24). Accordingly, the AVM proposed in (4) indicates that definiteness is a non-optional operator in English RPs, which can have two values: either definite or indefinite. As we will see in section 5, these values can be realized by means of the function words corresponding to the articles *the* or *a* or by demonstratives.

The AVMs that formalize the grammatical categories, which in RRG are part of the operator projection, are included in different ways in the GDE in ARTEMIS. On the one hand, their information can become part of the constituent projection and be integrated in the syntactic rules that describe each of the nodes of the layered structure of the clause (LSC) or the layered structure of the referential phrase (LSRP). The nodes are, therefore, “interpreted as a feature complex including different types of morphosyntactic parameters which are described as attributes” (Mairal-Usón and Cortés-Rodríguez 2017: 69). Notice, for instance, how the attribute illocutionary force (illoc) in example (5) and definiteness (def) in (6) are included on the list of attributes of the clause (CL) and the RP nodes, respectively.

(5) CL [akt=?, concept=?, emph= ?, *illoc*= ?, status= ?, t=?, tpl=?]

(6) RP [case=?, cnt= ?, concept=?, *def*= ?, dei= ?, n=?, p=?]

The grammatical information contained in the operators may, on the other hand, be conveyed by one or more function words. As stated before, it is vital, then, to provide the GDE in ARTEMIS with an exhaustive catalogue to list and describe them systematically. This process of including the function words relevant to ASD-STE100 in the catalogue of POS requires a consistent procedure. A list of approved functional items, together with their accepted forms in the ASD-STE100 dictionary, needs to be compiled, categorized and codified. We can illustrate this process by means of the function words *the* and *must*. In the case of the former, it is one of the two realizations of the category *article*, which has been tagged as ART, and whose attributes have to be listed in an AVM, see (7). In turn, each of these attributes needs to be provided with a description of the possible values they may present, as shown in (8) for definiteness.

(7) ART (article)

```
<Category Type=”ART”>
    <Attribute ID=”Countability”/>
    <Attribute ID=”Definiteness”/>
</Category>
```

(8) Definiteness

```
<Attribute ID=”Definiteness” obl=”+”num= “1” />
     <Value=?def </Value>
     <Value Tag=”definite”>d</Value>
     <Value Tag=”indefinite”>i</Value>
</Attribute>
```

Once the attributes and values have been formalized, a lexical rule that specifies the characteristics of each of the possible realizations of the category must be designed. In (9) we reproduce the specific lexical rule which describes the definite article *the*. It reads as follows: *the* is a definite (def=d) article (ART) and it can present one of the two possible values of countability (countable or uncountable):

(9 *the*: ART[cnt=c|u, def=d]

Another example could be the auxiliary *must*, a realization of the category deontic modal auxiliary that has been tagged as MODD. Following the same procedure, in (10) we can see the AVM which describes its attributes: the attribute illocutionary force is relevant for modal auxiliaries because they occupy the first slot of the nucleus node, a position which marks illocutionary force; the attribute modality distinguishes between the different modal meanings (ability, obligation or permission); finally, the attribute polarity allows the parser to differentiate between *can* and *cannot.*

```
(10) MODD (deontic modal)
     <Category Type="MODD">
          <Attribute ID="Illocutionary Force"/>
          <Attribute ID="Modality"/>
          <Attribute ID="Polarity"/>
     </Category>
```

A further description of the different values that characterize each of these attributes is needed, as illustrated in (11) for modality.

```
(11) Modality
     <Attribute ID="Modality" obl="*" num="1">
          <Value>?mod</Value>
          <Value Tag="ability">abl</Value>
          <Value Tag="obligation">obl</Value>
          <Value Tag="permission">perm</Value>
     </Attribute>
```

Finally, example (12) shows the lexical rule which accounts for the deontic modal *must,* one of the possible realizations of the category MODD.

(12) *must*: MODD [illoc=dec, mod=obl]

Lexical rules for function words may contain values which define operators pertaining to different levels, both of the LSC or the LSRP, as evidenced by the AVM for MODD above, which displays values associated with operators that belong to different levels of the LSC: the Clause level operator illocutionary force and the Core operators modality and polarity. Similarly, the description of the category *article* (ART) in (7) requires the combination of operators from the three different levels of the LSRP: the NuclearRP operator nominal aspect, the CoreRP operator number and the RP operator definiteness. The information contained in the lexical rules will always undergo a feature unification process. Unification can take place in two directions: vertically, when a feature percolates up to higher nodes in the layered structure, and, horizontally, when it involves matching values among constituents. These processes are illustrated in figure 5 for the demonstrative determiner *these*.

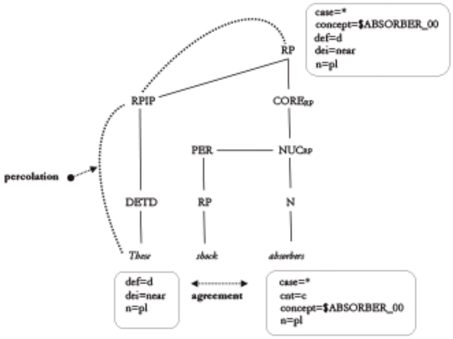


Figure 5. Feature unification processes in the RP: percolation and agreement.

4. THE CLAUSE IN ASD-STE100: IMPLICATIONS FOR POS

One of the most relevant categories for the description of the clause in ASD-STE100 is that of auxiliary verbs, instantiated both as primary or modal auxiliaries. As Van Valin and LaPolla (1997: 40) point out, "often in English and other Indoeuropean languages operators are coded on or as auxiliary verbs", hence the importance of providing an exhaustive description of this category. In what follows we will, therefore, offer the AVMs and lexical rules which formalize the characteristics of primary auxiliaries (AUX) and modal auxiliaries (MODD, MODST) in this controlled language.

The simplification that characterizes ASD-STE100 is strongly related to the nature of the documentation it deals with, mainly maintenance manuals in which communicative functions are restricted to giving instructions and describing procedures. This has a direct influence both in the syntax of the clauses and in the functional elements necessary to fulfill these communicative functions. An outstanding characteristic of ASD-STE100 clauses is the lack of interrogative structures, which entails that the clause operator illocutionary force can only present two values: declarative or imperative. This brings about a simplification of the syntactic rules which describe the clause and also a reduction in the ASD-STE100 catalogue of POS. Thus, the syntactic rules for the interrogative clause for natural English in ARTEMIS, as well as the description of the function words associated with interrogative clauses (Martín Díaz 2017) become redundant. Accordingly, the categories interrogative pronoun (*what*, *which*, *who*, *whom*, *whose*), interrogative adverb (*where*, *when*, *how* and *why*) and interrogative determiner (*whose, what and which)*, together with their corresponding tags (PROI, ADVI and DETI), disappear from the catalogue. Another consequence of the non-existence of interrogative clauses is the simplification in the description of the category primary auxiliary (AUX) in ASD-STE100, which now presents only two values for the attribute illocutionary force (declarative or imperative), as the lexical rule in (13) shows.

(13) AUX
[illoc: dec imp, emph=e null, per= 1 2 3, syn= APAR NEG VERB, tense= past pres fut]

The attribute emphasis (emph) that has been added to the description of AUX in (13) revealed necessary to describe emphatic sentences with *do*. Although emphasis is not considered an operator in RRG, it had to be included as a value of AUX to account for the instances of this type of clauses in the corpus analyzed[5]. As was the case with illocutionary force, the influence of emphasis starts in the NUC node and percolates to the Clause node. Examples (14) and (15) provide, respectively, a sentence from the corpus and the lexical rule which corresponds to the specific use of emphatic *do* in such a sentence.

(14) Be careful when you *do* work near the kneeling-manifold accumulator

(15) *do:* AUX [emph=e, illoc=dec, num=sg, per=2, syn=VERB, t=pres]

Another restriction that ASD-STE100 imposes on the category AUX at clause level concerns the formation of negative clauses. The ASD-STE100 specification manual rules against "contractions to make sentences shorter" (2017:1-4-2), therefore, enclitic negation is not permitted and AUX has to combine with the negative element *not* in the corresponding syntactic rule. In (16) we offer a corpus example of one such negative sentence, while (17) shows the lexical rule that accounts for the form of the periphrastic auxiliary *do* in the example.

(16) It is possible that the nose gear doors *do not* fully open because of their weight.

(17) *do:* AUX [illoc=dec, num=pl, per=3, syn=NEG, t=pres]

[5] The examples belong to a collected corpus of texts from aircraft maintenance instructions, courtesy of Airbus Seville.

The function word *not*, which is described in its corresponding lexical rule (19), is a realization of the category NEG described in (18).

```
(18) NEG:  <Attribute ID=”pol” obl=”*” num=”1”>
               <Value Tag=”negative”>neg</Value>
           </Attribute>
```

(19) *not:* NEG[pol: neg]

The combination AUX plus NEG is the instantiation of the LSC Core operator negation and is reflected in the lexical rule for the RP (13) as one of the three possibilities of syntactic agreement for AUX (syn= APAR|NEG|VERB). The attribute syn is also responsible for the difference between the periphrastic and emphatic uses of *do*: while periphrastic *do* collocates with NEG to form negative sentences, emphatic *do* can only collocate with VERB (bare infinitive form of the verb). The third value of syn, APAR (verb in the participle form), accounts for the passive uses of the auxiliary *be*. To the list of auxiliary verbs in ASD-STE100, and contrary to previous descriptions of the category AUX for ASD-STE100 (Martín-Díaz 2017; Fumero-Pérez 2018), *be* had to be included as a helping verb to account for the instances of passive voice in the corpus. Although the manual encourages the use of the active voice, in fact, in the texts there are many examples of passive structures (20) which cannot be overlooked, as they would imply a problem for the parsing procedure. To account for these instances of passive voice in the corpus, our proposal is to add to the AUX category the primary auxiliary *be* with a passive meaning exclusively. Such passive use of AUX is illustrated by the corpus example (20) and the corresponding lexical rule for this specific instance in (21):

(20) This mechanical deformation *is measured* by the strain gauges.

(21) *Is* : AUX [illoc=dec, num=sg, per= 3, syn=APAR, t=pres]

Including the passive use of AUX only implied updating the AVM of this category by adding the value APAR (verb in the participle form) to the attribute syn. To be able to formalize negative passive sentences[6], we only had to modify the attributes of the AUX category in the syntactic rule for the Nucleus node (see appendix I) as described by Cortés-Rodríguez and Mairal-Usón (2016: 17) and updated for ASD-STE100 by Díaz-Galán (2018: 92).

[6] It is interesting to notice that, when dealing with passive sentences, the parser will have to account for a “missing” argument in the verbal structure as it would do with other argumental constructions such as the inchoative. The information related to constructional meaning can be found in the Constructicon in ARTEMIS -the module dedicated to constructions.

Due to a further restriction prescribed by the writing manual on complex verb structures, which rules out aspectual distinctions, progressive *be* and perfective *have* become redundant. This implies that the nuclear operator aspect reveals unnecessary in ASD-STE100 and, therefore, it does not appear as one of the attributes of AUX.

Tense is the last attribute that characterizes primary auxiliaries. As can be seen in rule (13) above, it presents three values: past, present and future. The latter had to be incorporated to be able to account for the auxiliary *will*. This implies a modification of the values assigned to the clausal operator tense in RRG, in order to include future as a third indicator which situates the proposition temporally (Díaz-Galán 2018: 87). Differently from the case of *do*, in which the attribute syn distinguishes between its periphrastic and emphatic use (depending on whether it collocates with NEG or with VERB), in the case of *will*, syn is not a necessary feature since it would not have any distinctive value. This can be seen in the lexical rule (23):

(22) This will not have an unwanted effect on the parking brake.

(23) *will:* AUX [illoc=dec, t=fut]

The following table shows the description of the function words which instantiate the AUX category in ASD-STE100.

Modal auxiliaries are the means of coding two other operators of the LSC distinguished by RRG, namely, deontic modality (mod) and status (sta). In ASD-STE100 they are expressed by means of a limited number of modal auxiliaries. The deontic modals *can, cannot* and *must* are used to codify the LSC Core operator modality (permission, ability or obligation), as we can see in the following corpus examples:

(24) If you do, you *can* drain the fluid from the related hydraulic reservoir. (perm)

(25) If you *cannot* lift the aircraft, do the operational test of the Emergency Extension […]. (abl)

(26) If these parts are missing, you *must* send the equipment to the maintenance shop for an adjustment. (obl)

Epistemic *can* and *could* are responsible for encoding the Clause operator status, which in ASD-STE100 can only express possibility:

(27) Contamination of the tires by hydrocarbons *can* cause deterioration of the rubber. (poss)

Table 1. AUX category. Description and realizations in ASD-STE 100.

<table>
<tr><th colspan="2">AUX (primary auxiliaries) in ASD-STE100
[illoc: dec|imp, emph=e|null, per=1|2|3, syn=APAR|NEG|VERB, tense=pres|past|fut]</th></tr>
<tr><td>Periphrastic do</td><td><i>do not:</i> AUX [illoc= dec|imp, num= pl|sg, per= 1|2, syn= NEG, t= pres]
<i>does not:</i> AUX [illoc= dec|imp, num=sg, per= 3, syn= NEG, t= pres]
<i>do not:</i> AUX [illoc= dec|imp, num=pl, per= 3, syn= NEG, t= pres]
<i>did not:</i> AUX [illoc= dec|imp, syn=NEG, t= past]</td></tr>
<tr><td>Emphatic do</td><td><i>do:</i> AUX [emph= e, illoc= dec|imp , num= pl|sg, per=1|2, syn= VERB, t=pres]
<i>does:</i> AUX [emph= e, illoc= dec, num= sg, per=3, syn= VERB, t= pres]
<i>did:</i> AUX [emph= e, illoc= dec, num= pl |sg , per=1|2|3, syn=VERB, t=past]</td></tr>
<tr><td>Future Periphrastic will</td><td><i>will:</i> AUX [illoc= dec, syn=VERB, t= fut]
<i>will not:</i> AUX [illoc= dec, syn=NEG, t= fut]</td></tr>
<tr><td>Passive be</td><td><i>is:</i> AUX [illoc= dec, num=sg, per=3, syn=APAR, t=pres]
<i>are:</i> AUX [illoc= dec, num=sg, per=2, syn=APAR, t=pres]
<i>are:</i> AUX [illoc= dec, num=pl, per=3, syn=APAR, t=pres]
<i>was:</i> AUX [illoc= dec, num=sg, per=3, syn=APAR, t=past]
<i>were:</i> AUX [illoc= dec, num=pl, per=1|2|3, syn=APAR, t=past]</td></tr>
</table>

These two categories have been labelled MODD (deontic modality) and MODST (epistemic modality). The following tables set out the attributes and the possible values of the function words which integrate them.

Table 2. MODD category. Description and realizations in ASD-STE100.

<table>
<tr><th colspan="2">MODD (deontic modality) in ASD-STE100
[illoc=dec, mod=abl|obl|perm, pol=pos |neg]</th></tr>
<tr><td><i>can:</i></td><td>MODD [illoc=dec, mod=abl|perm]</td></tr>
<tr><td><i>cannot:</i></td><td>MODD [illoc=dec, mod=abl|perm, pol=neg]</td></tr>
<tr><td><i>must:</i></td><td>MODD [illoc=dec, mod=obl]</td></tr>
</table>

Table 3. MODST category. Description and realizations in ASD-STE100.

<table>
<tr><th colspan="2">MODST (epistemic modality) in ASD-STE100
[illoc=dec, sta=poss, t=pres|past]</th></tr>
<tr><td><i>can:</i></td><td>MODST [illoc=dec, sta=poss, t=pres]</td></tr>
<tr><td><i>could:</i></td><td>MODST [illoc= dec, sta=poss, t=past]</td></tr>
</table>

The auxiliary verbs we have described and formalized in this section, when present, are responsible for the codification of the information contained in the three levels of the clause operator projection relevant for ASD-STE100. While, as mentioned, the Nuclear operator aspect is redundant, at Core level, auxiliaries indicate modality and/or negation, whereas, at clause level, they can express illocutionary force, status and tense. In the absence of auxiliary verbs, the attributes encoding operator values are part of the information captured in the AVM of the predicate.

5. THE REFERENTIAL PHRASE IN ASD-STE100: IMPLICATIONS FOR POS

Whereas the simplification which supposedly characterizes the controlled language ASD-STE100 may be true for the clause, what we observe with respect to the phrase is a greater complexity, which results in the intricacy of lexical units and their phrasal projections in ASD-STE100 (Cortés-Rodríguez and Rodríguez-Juárez 2018). As mentioned in the introduction, the ASD-STE100 specification includes a dictionary of approved words which contains open and closed word classes. The former consists of a restricted set of nouns, verbs, adjectives and adverbs, which are complemented by the technical vocabulary specific to each manufacturer. The latter comprises the list of approved function words, which need to be tagged and described by means of lexical rules. In this section we will describe the most significant POS for the parsing of RPs in ASD-STE100.

As was the case with the clause, RPs also present a layered structure and operators which modify its different nodes. According to Van Valin (1997: 56), the operator that has scope over the NucleusRP node is nominal aspect (count-mass distinction); quantification and negation are the operators that modify the CoreRP node, and, finally, the RP operators are definiteness and deixis. Often, function words such as articles, determiners or quantifiers are responsible for the codification of the information signalled by these operators. Accordingly, in what follows the categories relevant to ASD-STE100 will be described. As with the lexical rules for function words related to the clause, the lexical rules which describe the function words relevant for the RP in ASD-STE100 may present values which define operators belonging to different levels of the LSRP. The first of these categories, the article (ART), realized either as definite or indefinite, is one of the most relevant categories that need to be defined in relation to the RP in ASD-STE100. Operators from two different levels of the LSRP merge in its description: countability (NuclearRP operator) and definiteness (RP operator), as the AVMs and lexical rules in table 4 illustrate.

Another salient category for the description of ASD-STE100 is that of demonstrative determiner (DETD), although only two forms are allowed: *this* and *these*. Their description (table 5) presents values which correspond to RP level operators, those which situate the phrase referentially (deixis) and with respect to definiteness. These operators, as Van Valin and LaPolla (1997: 758) indicate, have a discursive or pragmatic nature and, when present, are the outermost constituents in the RP affecting the RP as a whole. The category DETD also presents an attribute of number, although this CoreRP operator modifies the head noun of the RP, as a result of the unification of attributes, it also has to appear in the lexical rule in order to account for DETD-head agreement (see figure 5 above). Table 5 presents the description of this category and its lexical realizations.

Table 4. ART category. Description and realizations in ASD-STE100.

<table>
<tr><th colspan="3">ART (article) in ASD_STE100
[cnt=c|u, def= d|i]</th></tr>
<tr>
<td><Category Type=”ART”>
<AttributeID=”Count”/>
<Attribute ID=”Def”/>
</Category></td>
<td><Attribute ID=”Count” obl=”*” num=”s”>
<Value>?cnt</Value>
<Value Tag=”countable”>c</Value>
<Value Tag=”uncountable”>u </Value>
</Attribute>

<Attribute ID=”Def” obl=”+” num=”1”>
<Value>?def</Value>
<Value Tag=”definite”>d</Value>
<Value Tag=”indefinite”>i</Value>
</Attribute></td>
<td><i>a:</i>
ART [cnt=c, def=i]

<i>an:</i>
ART [cnt=c, def=i]

<i>the:</i>
ART[cnt=c|u, def=d]</td>
</tr>
</table>

Table 5. DETD category. Description and realizations in ASD-STE 100.

DETD (demonstrative determiner) in ASD-STE100 **[def=d|i, dei=near|far, n=pl|sg]**		

<Category Type=”DETD”>
<Attribute ID=”def”/>
<Attribute ID=”dei”/>
<Attribute ID=”n”/>
</Category>
	<Attribute ID= “Def” obl=”+” num=”1”>
	<Value>?def</Value>
	<Value Tag=”definite”>d</Value>
	<Value Tag=”indefinite”>i</Value>
	</Attribute>
	
	<Attribute ID=”Dei” obl=”+”num=”1”>
	<Value>?dei</Value>
	<Value>near</Value>
	<Value>far</Value>
	</Attribute>
	
	<Attribute ID=”Num” obl=”+” num=”1”>
	<Value>?n</Value>
	<Value Tag=”plural”>pl</Value>
	<Value Tag=”singular”>sg</Value>
	</Attribute>
		this:
		DETD[def=d, dei=near, n=sg]
		
		these:
		DETD[def=d, dei=near, n=pl]

Another possible realization of the category determiner is that of possessive determiner (DETP), of which we may find only two forms in ASD-STE100, namely *its* and *their*. Their behaviour is exactly the same as that of the demonstrative determiner; they occupy the RP initial position and have the whole phrase in their scope. The attributes which describe this category are the same as those for DETD, with the exception of deixis, which is the defining attribute of demonstratives. Table 6 lists the words attributes and values of this category, together with the lexical rules of the two approved function words.

Table 6. DETP category. Description and realizations in ASD-STE 100.

DETP (possessive determiner) in ASD-STE100 **[def=d|i, n=pl|sg]**		

<Category Type=”DETP”>
<Attribute ID=”Def”/>
<Attribute ID=”Num”/>
</Category>
	<Attribute ID= “Def” obl=”+” num=”1”
	<Value>?def</Value>
	<Value Tag=”definite”>d</Value>
	<Value Tag=”indefinite”>i</Value>
	</Attribute>
	
	<Attribute ID=”Num” obl=”+”num=”s”>
	<Value>?n</Value>
	<Value Tag=”plural”>pl</Value>
	<Value Tag=”singular”>sg</Value>
	</Attribute>
		Its:
		DETP[def= d, n=sg]
		
		their:
		DETP[def= d, n=pl]

The CoreRP operator quantification situates the RP with respect to quantity, within the RP this function is realized by means of quantifying determiners (DETQ). The attributes that define this category have been described by Cortés-Rodríguez (2016: 102) and include the following values: countability, number, quantification and polarity. The author considers that the two CoreRP operators negation and quantification merge, and proposes that they should be expressed as a set of values which range from absolute positive (*all)* to absolute negative (*no*), with two intermediate values: relative positive (*many*) and relative negative *(few*). In line with the previous description, we have designed the lexical rules for the function words that integrate this category in ASD-STE100 and that we list in table 7 below:

Table 7. DETQ category. Description and realizations in ASD-STE 100.

DETQ (quantifying determiner) in ASD-STE100 **[cnt=c|u, n=pl|sg, quant=an|ap|qa|rn|rp]**		

Category Type="DETQ">
<Attribute ID="Count"/>
<Attribute ID="Num"/>
<Attribute ID="Quant"/>
</Category>
	Attribute ID="Count"obl="*"num="s">
	<Value>?cnt</Value>
	<Value Tag="countable">c</Value>
	<Value Tag="uncountable">u</Value>
	</Attribute>
	
	<Attribute ID="Num" obl="+"num="s">
	<Value>?n</Value>
	<Value Tag="plural">pl</Value>
	<Value Tag="singular">sg</Value>
	</Attribute>
	
	<Attribute ID="Quant" obl="*"num="1">
	<Value>?quant</Value>
	<ValueTag="absolute negative">an</Value>
	<ValueTag="absolute positive">ap</Value>
	<ValueTag="quasi absolute positive">qa</Value>
	<ValueTag="relative negative">rn</Value>
	<ValueTag="relative positive">rp</Value>
	</Attribute>
		all:
		DETQ[cnt=c|u,n=pl,quant=ap]
		
		many:
		DETQ[cnt=c, n=pl, quant=rp]
		
		no:
		DETQ[cnt=c|u,n=pl|sg,quant=an]
		
		most:
		DETQ[cnt=c, n= pl , quant=rp]
		
		much:
		DETQ[cnt=u, n=sg, quant= rp]
		
		each:
		DETQ[cnt=c, n=sg, quant=ap]

The CoreRP operator number is present as an attribute in many of the lexical rules described, but the only case in which this operator instantiates in a POS is in the category numeral with the function determiner, which we have tagged as DETN. Table 8 presents a sample of the values and attributes that characterize cardinal numerals (DETNC) and ordinal numerals (DETNO) and the lexical rules for the specific realization of two members of each category.

In this section, we have presented the codification of the information contained in the most relevant categories for the description of RPs in ASD-STE100. The representation of the corresponding function words relate to operators belonging to the three levels of the LSRP. Although there is a reduction of the possible realizations of the function words in ASD-STE100, this does not seem to influence the values of the operator projection of the LSRP.

Table 8. Numerals. Description and realizations in ASD-STE 100.

DETNC (cardinal numbers) in ASD-STE100	
Category Type=”DETNC”> <Attribute ID=”Num”/> </Category>	*One:* DETNC [n=s] *Five:* DETNC [n=pl]
DETNO (ordinal numbers) in ASD-STE100	
Category Type=”DETNO”> <Attribute ID=”Num”/> </Category>	*First:* DETNO [n=pl|sg] *Fifth:* DETNO [n=pl]

6. CONCLUSION

While in ARTEMIS, for parsing purposes, lexical content can be retrieved either from the FunGramKB Core Ontology or from the Satellite Ontology, there is a need for a specific catalogue of function words for ASD-STE100 tailored to the restrictions imposed on this controlled language. Within ARTEMIS, function words are stored in the GDE and described as lexical rules, one of the three possible types of rules in such component together with syntactic and constructional rules. Once available in the GDE, function words can be inserted in the syntactic rules, both at clausal or phrasal level. Function words prove fundamental for processing language since they instantiate the realizational possibilities of the operator projection in RRG, both at clause and phrase level. Given that in the ARTEMIS GDE there is no operator projection -as it follows the paradigm of unification grammars- the information provided by operators has to be formalized as AVMs. As we have seen, such descriptive devices, which characterize a given category by means of different attributes and their corresponding values, can be a) integrated or unified with syntactic rules or b) used to create the lexical rules which describe function words.

In the process of creating a catalogue of function words in ARTEMIS for ASD-STE100, we have designed the primary lexical rules necessary to parse the simple sentence and the RP in this controlled language. Further research would be necessary to account for the complex sentence. Nevertheless, the values of the categories and the whole range of functionally motivated AVMs designed for the description of the function words presented here make an important contribution to the characterization of function words, as some of their properties are reflected on the constituent projection both of the LSC and the LSRP, assessing, therefore, the suitability of RRG for NLP tasks.

REFERENCES

AeroSpace and Defence Industries Association of Europe. 2017. *International Specification for the Preparation of Technical Documentation in a Controlled Language* 7 (january).

Ball, J. 2017. "Using NLU In Context For Question Answering: Improving on Facebook's BABI Tasks." <https://arxiv.org/ftp/arxiv/papers/1709/1709.04558.pdf>. (Accessed 24 April 2019)

Cortés-Rodríguez, F. 2016. "Towards the computational implementation of RRG". *Círculo de Lingüística Aplicada a la Comunicación* 65: 75-108.

Cortés-Rodríguez, F. and R. Mairal-Usón. 2016. "Building an RRG computational grammar". *Onomazein* 34: 86-117.

Cortés-Rodríguez, F. and C. Rodríguez-Juárez. 2018. "Parsing Phrasal Constituents in ASD-STE with ARTEMIS". *Voprosy Kognitivnoy Lingvistiki* 3: 97-109.

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.

Díaz-Galán, A. and M. Fumero-Pérez. 2016. "Developing parsing rules within ARTEMIS: The case of Do auxiliary insertion". *Understanding meaning and knowledge representation: From theoretical and cognitive linguistics to natural language processing.* Eds. C. Periñán-Pascual and E. Mestre-Mestre. Cambridge: Cambridge Scholars Publishing. 283-302.

Diedrichsen, E. 2014. "A Role and Reference Grammar Parser for German". *Language Processing and Grammars. The role of functionally oriented computational* *models*. Eds. B. Nolan and C. Periñan- Pascual. Amsterdam / Philadelphia: John Benjamins. 105-142.

Fumero-Pérez, M. 2018. "Adapting ARTEMIS to Parsing a Controlled Language". *Voprosy Kognitivnoy Lingvistiki* 3: 71-82.

Fumero-Pérez, M. and A. Díaz-Galán. 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: 33-44.

Goldberg, A. 1995. *Constructions: A construction grammar approach to argument structure*. Chicago: University of Chicago Press

Guest, E. 2008. "Parsing for Role and Reference Grammar". *Investigations of the Syntax-Semantics-Pragmatics Interface*. Ed. R. Van Valin. Amsterdam: John Benjamins. 435-454.

Mairal Usón, R. and F. Cortés Rodríguez. 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.

Mairal Usón, R. and F. Ruiz de Mendoza. 2009. "Levels of description and explanation in meaning construction". *Deconstructing Constructions*. Eds. C. Butler and J. Martín Arista. Amsterdam: John Benjamins. 53-98.

Martín-Díaz, M. 2017. "An account of English YES /No interrogative sentences within ARTEMIS". *Revista de Lenguas Para Fines Específicos* 23 (1): 421-62.

Osswald, R. and L. Kallmeyer. Forthcoming. "Towards a formalization of Role and Reference Grammar". To appear in *Applying and Expanding Role and Reference Grammar*. Eds. R. Kailuweit, E. Staudinger and L. Künkel. Freiburg University Press. <http://user.phil.hhu.de/~osswald/publications/osswald_kallmeyer-rrg_formalization.pdf>. (Accessed 10 October 2018)

Periñán-Pascual, C. 2012. "En defensa del procesamiento del lenguaje natural fundamentado en la lingüística teórica". *Onomázein* 26: 13-48.

Periñán-Pascual, C. 2013. "Towards a Model of Constructional Meaning for natural Language Understanding". *Linking Constructions into Functional Linguistics: The Role of Constructions in RRG Grammars (Studies in Language Series)*. Eds. B. Nolan and E. Diedrichsen. Amsterdam/Philadelphia: John Benjamins. 205-230.

Periñán-Pascual, C. and F. Arcas-Túnez. 2010. "The Architecture of FungramKB." *Proceedings of the 7th International Conference on Language Resources and Evaluation*. Eds. N. Calzolari, K. Choukri, B. Maegaard, J. Mariani, J. Odijk, S. Piperidis, M. Rosner and D. Tapias. Malta: European Language Resources Association. 2667-2674.

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

Periñán-Pascual, C. and R. Mairal Usón. 2010a. "La gramática de COREL: un lenguaje de representación conceptual". *Onomázein* 21: 11-45.

Periñán-Pascual, C. and R. Mairal Usón. 2010b "Enhancing UniArab with FunGramKB". *Procesamiento del Lenguaje Natural* 44: 19-26.

Rodríguez–Juárez, C. 2017. "On the computational treatment of constructions: the place of locative constructions in a knowledge base". *Onomázein* 21: 11-45.

Ruiz de Mendoza, F. and R. Mairal Usón. 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. and E. Bender. 2003. *Syntactic Theory: Formal Introduction*. Stanford: Center for the Study of Language and Information.

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

Van Valin, R. D. 2008. "RPs and the nature of lexical and syntactic categories in Role and Reference Grammar". *Investigations of the Syntax-Semantics-Pragmatics Interface*. Ed. R. D. Van Valin. Amsterdam/Philadelphia: John Benjamins. 161-178.

Van Valin, R. D. and R. LaPolla. 1997. *Syntax*. Cambridge: Cambridge University Press.

APPENDIX I

Simplified rule for the LSC Nucleus node in ASD-STE100:

> NUCàPRED ‖ AUX PRED ‖ AUX NEG PRED ‖ MODD PRED ‖ MODD NEG PRED ‖ MODST PRED ‖ MODST NEG PRED

Examples of the possible realization of the clausal nucleus (NUC) node according to the previous rule:

> NUCà PRED: eg. install ‖
> NUCàAUX PRED: eg. is installed (passive) /do install (emphatic) ‖
> NUCàAUX NEG PRED: eg. is not installed (passive)/ do not install (periphrastic) ‖
> NUCàMODD PRED: eg. must install ‖
> NUCàMODD NEG PRED: eg. must not install ‖
> NUCàMODST PRED: eg. could install ‖
> NUCàMODST NEG PRED: eg. could not install

Updated rule for the NUC node containing attributes:

> NUCà PRED [concept= ?, illoc=?, num=?, recip=?, reflex=?,per=?, tpl=?, t=?] ‖ AUX [illoc=dec imp, emph=e null, per=1 2 3, syn=APAR NEG VERB, t=pres past fut] PRED [concept=?, illoc=?, num=?, recip=?, reflex=?,per=?, tpl=?, t=?] ‖ AUX [illoc=dec imp, emph=e null, per=1 2 3, syn=APAR NEG VERB, t=pres past fut] NEG [pol=neg] PRED [concept=?, illoc=?, num=?, recip=?, reflex=?,per=?, tpl=?, t=?] ‖ MODD [illoc=dec, mod= abl obl perm, pol= pos neg, syn= verb null, t=pres past fut] PRED [concept=?, illoc=?, num=?, recip=?, reflex=?,per=?, tpl=?, t=?] ‖ MODD [illoc=dec, mod= abl obl perm, pol= pos neg, syn= verb null, t=pres past fut null] NEG [pol=neg] PRED [concept=?, illoc=?, num=?, recip=?, reflex=?,per=?, tpl=?, t=?] ‖ MODST [illoc=dec, sta=poss, t=pres past] PRED [concept=?, illoc=?, num=?, recip=?, reflex=?,per=?, tpl=?, t=?] ‖ MODST [illoc=dec, sta=poss, t=pres past] NEG [pol=neg] PRED [concept=?, illoc=?, num=?, recip=?, reflex=?,per=?, tpl=?, t=?]

APPENDIX II

ADVI	Interrogative adverb
APAR	Verb in the past participle form
ART	Article
ARTEMIS	Automatically Representing Text Meaning via an Interlingua-based System
ASD-STE100	The Aerospace and Defence Industries Association of Europe Simplified Technical English
AUX	Primary Auxiliary
AVM	Attribute Value Matrix
CLS	Conceptual Logical Structure
COREL	Conceptual Representation Language
DETD	Demonstrative determiner
DETI	Interrogative Determiner
DETN	Ordinal numeral
DETNC	Cardinal numeral
DETP	Possessive Determiner
DETQ	Quantifying Determiner
FunGramKB	Functional Grammar Knowledge Base
GDE	Grammar Development Environment
L1-Construction	Level 1 (argumental) construction
LSC	Layered Structure of the Clause
LSRP	Layered Structure of the Referential Phrase
MODD	Deontic Modal Auxiliary
MODST	Epistemic Modal Auxiliary
MP	Modifier Phrase
NLP	Natural Language Processing
NLU	Natural Language Understanding
POS	Part of Speech
PROI	Interrogative Pronoun
RP	Referential Phrase
RRG	Role and Reference Grammar
UML	Unified Modelling Language