

Ontology Based Interlingua Translation

Leonardo Lesmo, Alessandro Mazzei, and Daniele P. Radicioni

University of Turin, Computer Science Department
Corso Svizzera 185, 10149 Turin
{lesmo,mazzei,radicion}@di.unito.it

Abstract. In this paper we describe an interlingua translation system from Italian to Italian Sign Language. The main components of this systems are a broad coverage dependency parser, an ontology based semantic interpreter and a grammar-based generator: we provide the description of the main features of these components.

1 Introduction

In this paper we describe some features of a system designed to translate from Italian to Italian Sign Language (henceforth LIS). Many approaches have been proposed for automatic translation, which require different kinds of linguistic analysis. For instance, the *direct* translation paradigm requires just morphological analysis of the source sentence, while the *transfer* translation paradigm requires syntactic (and sometimes semantic) analysis too [1]. In contrast, our architecture adheres to the *interlingua* translation paradigm, i.e. it performs a deep linguistic processing in each phase of the translation, i.e. (1) deep syntactic analysis of the Italian source sentence, (2) semantic interpretation, and (3) generation in LIS of the target LIS sentence. These three phases form a pipeline of processing: the syntactic tree produced in the first phase is the input for the second phase, i.e semantic interpretation; similarly, the semantic structure produced in the second phase is the input of the third phase, i.e. generation. In order to work properly, Interlingua pipeline requires good performances in each phase of the translation. Moreover, since the semantic interpretation is crucially related to the world knowledge, the state-of-the-art computational linguistic techniques allow the interlingua approach to work only on limited domain [1]. In our work, we concentrate on the classical domain of weather forecasts.

A challenging requirement of our project is related to the target language, the LIS, that does not have a *natural* written form (which is typical of the signed languages). In our project we developed an *artificial* written form for LIS: this written form encodes the main morphological features of the signs as well as a number of non-manual features, as the gaze or the tilt of the head. Anyway, for sake of clarity in this paper we report a LIS sentence just as a sequence of *GLOSSAS*, that is the sequence of the names¹ of the signs, without any extra-lexical feature.

¹ A name for a sign is just a *code* necessary to represent the sign. As it is customary in the sign languages literature, we use names for the signs that are related to their rough translation into another language, Italian in our work.

The building blocks of our architecture are the dependency parser for syntactic analysis, the ontology-based semantic interpreter, the CCG-based generator. In the paper, we first introduce the dependency parser (Section 2), then we focus on the description of the main issues of the semantic interpretation and provide a case study on ordinal numbers (Section 3). A key point in the semantic interpretation is that the syntax-semantics interface used in the analysis is based on an ontology, similar to [2]. The knowledge in ontology concerns the domain of application, i.e. weather forecasts, as well as more general information about the world. The latter information is used to compute the sentence meaning. The result of the semantic interpretation is a complex fragment of the ontology: predicate-argument structures and semantic roles describing the sentence are contained in this fragment. In Section 4 we describe the generation phase, and illustrate the a combinatory categorial grammar that we devised for LIS. Finally, in Section 5 we conclude the paper and point out some future developments to the system.

2 Syntactic Analysis

In limited domains (as the one of weather forecasts) it is possible to obtain a “deep understanding” of the meaning of texts. To get this result, we need the detailed syntactic structure of the input sentences and specific information about the meaning of the words appearing in the sentences. The syntactic structure is produced by the TULE parser [3]. It uses a morphological dictionary of Italian (about 25,000 lemmata) and a rule-based grammar. The final result is a “dependency tree”, that makes clear the structural syntactic relationships occurring between the words of the sentence. After two preliminary steps (the *morphological analysis* and *part of speech tagging*, necessary to recover the lemma and the part of speech (PoS) tag of the words), the sequence of words goes through three phases: *chuncking*, *coordination analysis*, and *verbal subcategorization*.

Let us consider the following sentence: “Locali addensamenti potranno interessare il settore nordorientale” (*Local cloudiness could concern the northeastern sector*). By looking for chunks (i.e. sequences of words usually concerning noun substructures), we get “Locali addensamenti” and “il settore nord-orientale”. Then verbal subcategorization is used to attach these chunks to the verbs “potere” and “interessare” and for inserting the trace. Each word in the sentence is associated with a node of the tree. Actually, the nodes include further data (e.g., the gender and number for nouns and adjectives and verb tenses) which do not appear in the figure for space reasons. The nodes are linked via labeled arcs that specify the role of the dependents with respect to their governor (the parent). For instance, “addensamento” (*cloudiness*) is the subject of the verb “potere” (to *can*: verb-subj), while “il” (*the*) is the direct object of “interessare” (to *interest*: verb-obj). In the Figure, there is also a special node (framed by a dashed line and labeled *t*), which is a “trace”. It specifies that the subject of “interessare” is “addensamento”, although the node associated with it syntactically depends on “potere”. In other words this node, which does not correspond to any word in the sentence, enables us to specify that “addensamento” is a subject shared by “potere” and “interessare”.

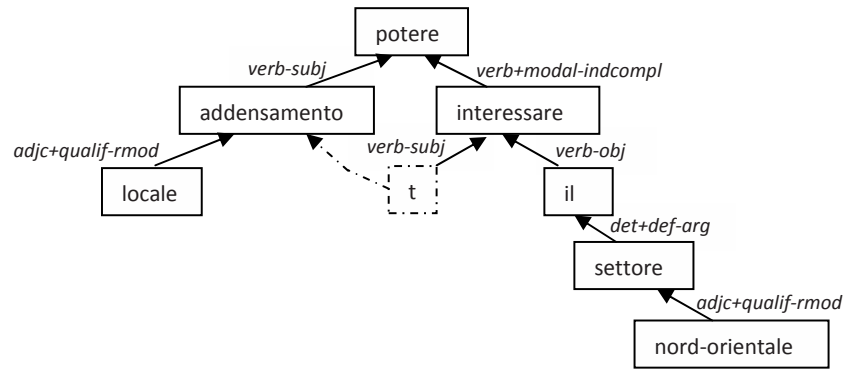


Fig. 1. Syntactic structure of the sentence “Locali addensamenti potranno interessare il settore nord-orientale” (*Local cloudiness could concern the north-eastern sector*)

3 Knowledge Representation and Semantic Interpretation

In the overall architecture of the system the ontology is accessed to build a semantic representation of the input sentence, which is then used by the generative process. Basically the system searches for a match between the (annotated) syntactic trees and the concepts in a domain ontology. We now introduce two preliminary issues, that are the notion of *semantic proximity* and the problem of *linguistic mediation*; then in Section 3.1 we describe the taxonomy of the entity *description*, and in 3.3 we provide an example to illustrate how the system copes with the linguistic phenomenon of ordinals; we then illustrate the use of the *ordinal-description* entity which is central to the interpretation process.

In the present setting we build on the notion of *semantic proximity* between concepts that are present in a given region. Intuitively, the proximity between two concepts can be defined as the number of intervening steps between them in a concept hierarchy [4]. The process of semantic interpretation can be cast to the problem of finding a path between pairs of words. A shortest path is searched that represents the strongest semantic connection between the words; and in turn, the strongest semantic connection is that minimizing the (semantic) distance between the considered words.

In general, an ontology can collect two kinds of rather different entities. On the one side entities that are concerned with the application domain, such as temporal entities and geographic regions, weather status. On the other side we deal with the *description* of such entities, which is rooted in the linguistic mediation that has to do with ‘talking of’ things, rather than with ‘things themselves’. Accordingly, in devising ontologies (and in particular ontologies of intrinsically linguistic and communicative acts, such as weather forecasts) one has to deal with two problems: how to represent the knowledge about the world, and how is that knowledge connected to language that is needed to talk about the knowledge about the world. This problem is sometimes referred to as *ontological stratification*, and it has received different answers in the scientific community [5]. A possible solution to the ontological stratification problem consists in considering multiple ontological levels, such as a *material* level of constitution and the ‘*objects themselves*’ level [6]. Under a different perspective, the dichotomy between

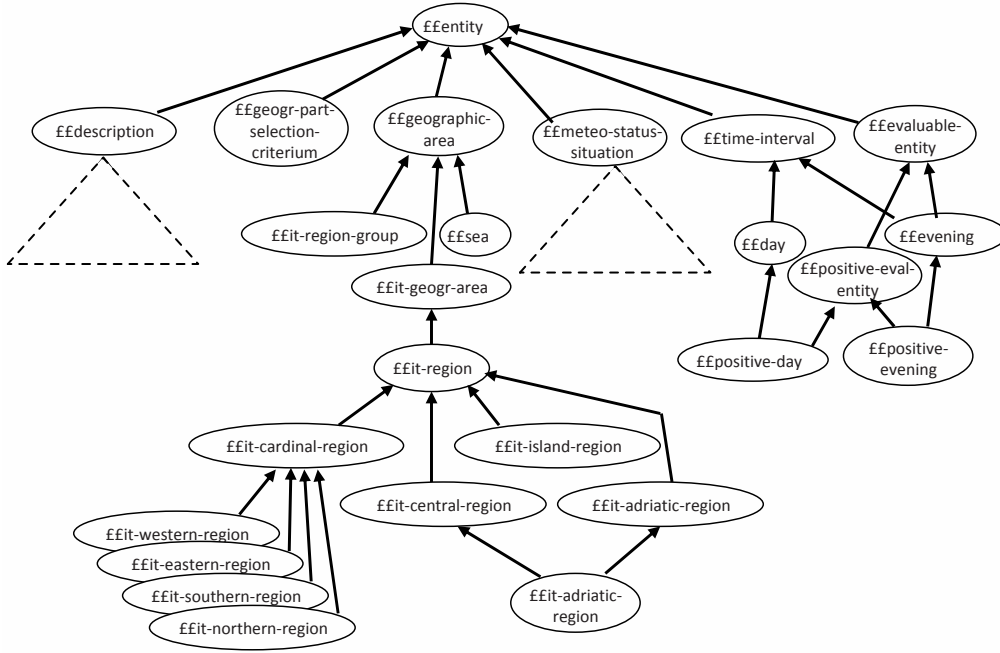


Fig. 2. The top level of the weather forecasts ontology. Dashed triangles represent regions of the hierarchy that are illustrated separately in Figure 3 and 4.

the world and its description inspired the so called *D&S* (so named after Descriptions and Situations) and its constructive counterpart *c.DnS* [7,8]. In particular, *c.DnS* can be used to extend the DOLCE foundational ontology [9] by providing it with an epistemological perspective “from which the entities of the domain are considered”. In particular, the mentioned approach proposes to describe *conceptualizations* from some given domain through *descriptions* and the settings (also called states of affairs) relevant to the considered domain through *situations*. In our ontology descriptions are entities separated from entities representing concepts themselves. For example, if today is October 29, 2010 in the ontology we distinguish the (deictic) description ‘today’, from the referred instance of day. Similarly ‘October 29, 2010’, would be represented like another (absolute) description of the same instance of day.

3.1 The Ontology

The top level of the ontology is illustrated in Figure 2.² In the following we denote *concepts* (classes) with the $\mathbb{L}\mathbb{L}$ prefix; *instances* have a \mathbb{L} prefix, and *relations* and their instances are prefixed with \mathbb{L} . We start by considering the classes most relevant to weather forecasts, that is $\mathbb{L}\mathbb{L}$ *meteo-status-situation*, $\mathbb{L}\mathbb{L}$ *geographic-area* and $\mathbb{L}\mathbb{L}$ *description*.

- $\mathbb{L}\mathbb{L}$ *meteo-status-situation*. This is the most relevant subclass in the present setting, since it refers about the possible weather situations, thus providing the starting point

² We defer to a future work the investigation of how the present ontology could be connected to a foundational ontology, such as DOLCE.

–in principle– to every weather forecast. It may concern the sea status (and the *££sea-current*), a generic weather status (in particular if it is stable or not) or possible atmospheric events such as snow, rain or clouds. Three subclasses are rooted in *££meteo-status-situations*: *££sea-status-situation*, *££weather-event* and *££weather-status-situation*.

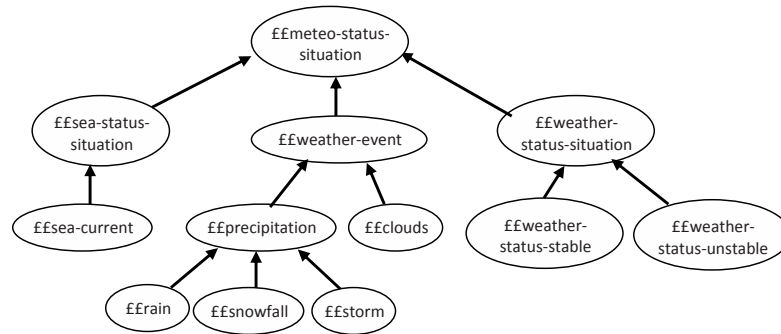


Fig. 3. The portion of the ontology describing weather situations

- *££time-interval*. Any weather situation holds in a specific temporal interval, thereby making time a fundamental element in weather forecasts. Such time interval could last one or more days or a part of a day.
- *££geographic-area*. Any weather situation holds in a specific place; in particular, the relevant places are geographic areas. A *££geographic-area* can be an Italian region, a group of regions, a sea, or may be identified by specifying a cardinal direction (North, South, ...).
- *££description*. In the hierarchy rooted in the concept *££description*, particular relevance have the deictic descriptions (see Figure 4), since most temporal descriptions (*today*, *tomorrow*, but also the weekday names, as *Monday*, *Tuesday*, ...) are deictic in nature.

Further relevant subclasses of *££entity* are *££degree*, which is used to specify, for instance, that the weather is more or less stable; *££reified-relation*, about which we elaborate in the following.

Relations. The last relevant portion of the ontology concerns *relations*. Although the ontology has no axioms, class concepts are connected through relevant relations. In turn, relations constitute the basic steps to form paths. All relations in the ontology are binary, so that the representation of relations of arity greater than 2 requires them to be reified. In Figure 5 we report two example relations that occur in the weather forecast domain. Relations are represented as arrows with small boxes. The *domain* of the relation is the node that the arrow leaves, while the *range* is the node that the arrow enters. The name of the relation is reported near the small box. The functionality information has the usual meaning, and is used to add constraints on the fillers of a relation with respect to some class. Namely, 1:1 means that both the relation and its inverse are *functional*; 1:N means that each individual of the domain can be associated with N individuals of the

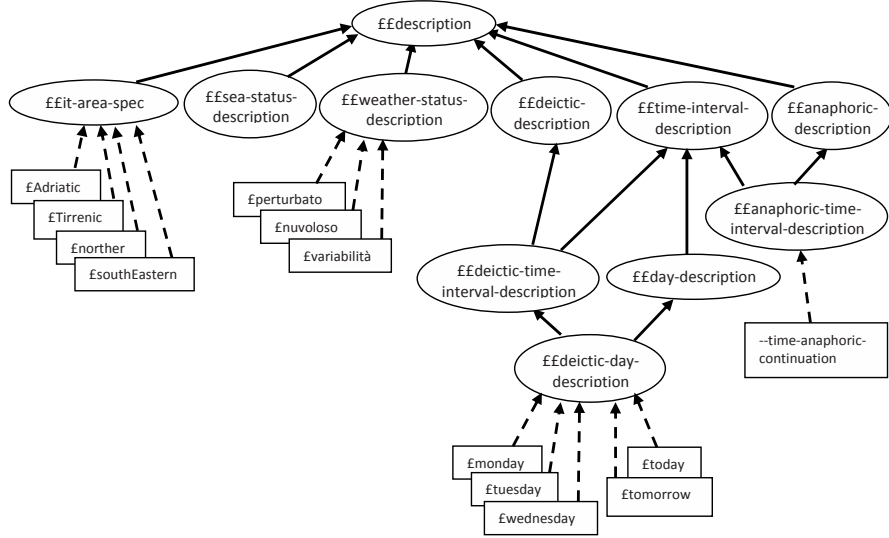


Fig. 4. The portion of the ontology concerning descriptions (with some example instances, depicted as boxes)

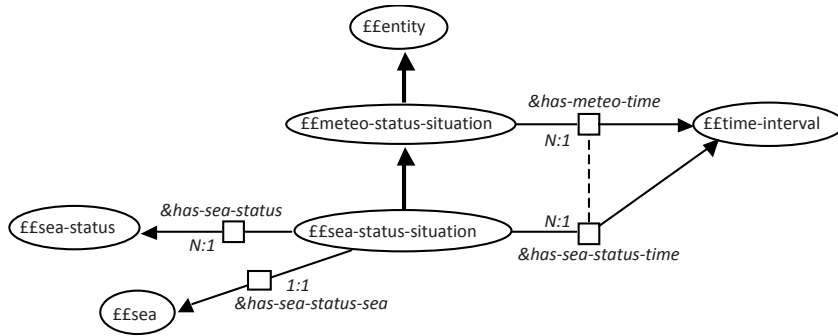


Fig. 5. Use of relations to represent the features of $££sea-status-situation$

range, but not viceversa. The converse is expressed by the notation N:1. N:M refers to the absence of functionality constraints. The dashed link connecting $&has-meteo-time$ and $&has-sea-status-time$ specifies that the latter relation restricts the former one.

3.2 Semantic Interpretation

The basic assumption underlying the semantic interpretation is that the meaning of words is expressed in terms of ontology nodes, and that a central component of the overall meaning of the sentence is a complex path on the ontology that we call *ontological restriction*. In this Section we define the *meaning interpretation function* $\mathcal{M}_{\mathcal{O}}$: we start from the dependency tree of the sentence, and on the basis of the lexical meaning of the words (given in terms of an ontology \mathcal{O}) we compute the ontological restriction.

Given a sentence S and the corresponding syntactic analysis expressed as a dependency tree $depTree(S)$, the meaning of S is computed by applying the meaning

interpretation function to the root of the tree, that is $\mathcal{M}_{\mathcal{O}}(\text{root}(\text{depTree}(S)))$. In procedural terms, the meaning corresponding to a sentence is computed in two steps: (i) we annotate each word of the input sentence with the corresponding lexical meaning; (ii) we build the actual ontological representation in a quasi-compositional way, by joining paths found in the ontology in a single representation which is a subgraph (with possible redundancies) of the ontology itself. These two steps can be formalized as a meaning interpretation function $\mathcal{M}_{\mathcal{O}}$ defined³ as:

$$\mathcal{M}_{\mathcal{O}}(n) := \begin{cases} \mathcal{LM}_{\mathcal{O}}(n), & \text{if } n \text{ is a leaf} \\ \dot{\cup}_{i=1}^k (\mathcal{CP}_{\mathcal{O}}(\mathcal{LM}_{\mathcal{O}}(n), \mathcal{M}_{\mathcal{O}}(d_i))), & \text{otherwise} \end{cases}$$

where n is a node of the dependency tree and d_1, d_2, \dots, d_k are its dependents. $\mathcal{LM}_{\mathcal{O}}(w)$ is a function that extracts the lexical meaning of a word w : that is, a class or an individual in the ontology \mathcal{O} . The meaning is determined by accessing the dictionary. $\mathcal{CP}_{\mathcal{O}}(y, z)$ is a function that returns the shortest path on \mathcal{O} connecting y to z . The search for connections is based on the idea that the shortest path that can be found in the ontology between two nodes represents the stronger semantic connection between them; consequently, such path must be used to build the semantic representation. Finally, the operator $\dot{\cup}$ is used to denote a particular merge operator. As a general strategy, shortest paths are composed with a union operator, but each $\mathcal{CP}_{\mathcal{O}}(y, z)$ conveys a set of ontological constraints: the merge operator takes all such constraints into account in order to build the overall complex ontological representation. A particular case of ontological constraints is present in the interpretation of ordinal numbers, which is discussed in next Section.

3.3 A Case Study: The Ordinal Numbers

In order to translate from Italian to LIS, we have to account for a number of semantic phenomena appearing in the particular domain chosen as pilot study, i.e. weather forecast. One of the most frequent constructions are ordinal numbers. Let us consider the simple phrase *l'ultimo giorno del mese* (*the last day of the month*). The (simplified) dependency structure corresponding to this phrase is depicted in Figure 6: the head word *giorno* (*day*) has two modifying dependents, *ultimo* (*last*) and *mese* (*month*). Since the

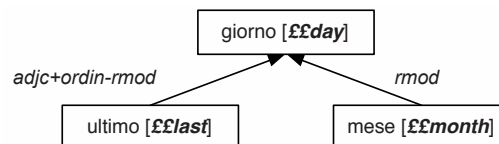


Fig. 6. The dependency analysis of *ultimo giorno del mese* (*last day of the month*) enriched with lexical meaning (in bold face)

³ For sake of simplicity in this definition we do not describe the mechanism used for ambiguity resolution.

interpretation relies heavily on the access to the ontology, we first describe the part of the ontology used for the interpretation and then we illustrate the application of the function \mathcal{M}_O on the given example.

The relevant fragment of the ontology is organized as shown in Figure 7, where it has been split in two parts. The upper part –labeled *TEMPORAL PARTS*– describes the reified $\mathbb{L}\mathbb{L}$ part-of relation and its temporally specialized subclasses. The lower part –labeled *ORDINALS*– is constituted by some classes that account just for ordinal numbers. In the *TEMPORAL PARTS* region of the Figure we find the $\mathbb{L}\mathbb{L}$ temporal-part-of (reified) sub-relation, which, in turn, subsumes $\mathbb{L}\mathbb{L}$ day-month-part-of. This specifies that days are parts of months, so that *day of the month* can be interpreted as *the day which is part of the month*. The $\mathbb{L}\mathbb{L}$ part-of relation has two roles: we use the term *role* to refer to the binary relation associated with a participant in a reified relation. These roles are “value-restricted” as $\&$ day-in-daymonth and $\&$ month-in-daymonth respectively, for what concerns $\mathbb{L}\mathbb{L}$ day-month-part-of. The most relevant class in the *ORDINALS* part of Figure 7 is the class $\mathbb{L}\mathbb{L}$ ordinal-description. It is the *domain* of three roles, 1) $\&$ ord-described-item, 2) $\&$ reference-sequence and 3) $\&$ ordinal-desc-selector. The range of the first relation $\&$ ord-described-item is the item whose position in the sequence is specified by the ordinal, that is a $\mathbb{L}\mathbb{L}$ sequenceable-entity. The range of the second relation $\&$ reference-sequence is the sequence inside which the position makes sense, that is an $\mathbb{L}\mathbb{L}$ entity-sequence. The range of the third relation $\&$ ordinal-desc-selector is item that specifies the position, that is a $\mathbb{L}\mathbb{L}$ ordinal-selector. In the example, \mathbb{L} last is an instance of $\mathbb{L}\mathbb{L}$ ordinal-selector. Of course, any (true) ordinal (first, second, thirtythird) can fill that role. The two portions of the ontology are connected by two arcs. The first arc specifies that a $\mathbb{L}\mathbb{L}$ time-interval is a subclass of $\mathbb{L}\mathbb{L}$ sequenceable-entity (so that one can say *the fourth minute, the first year*, and so on). The second arc specifies that $\mathbb{L}\mathbb{L}$ month is subclass of $\mathbb{L}\mathbb{L}$ day-sequence, which in turn is subclass of $\mathbb{L}\mathbb{L}$ entity-sequence. As a consequence it can play the role (can be *range*) of the $\&$ reference-sequence. Applying the meaning interpretation function to the considered example consists of three steps: 1. we compute the connection path (\mathcal{CP} function) between words *giorno* and *ultimo* (i.e., the first branch of the dependency tree in Figure 7) we obtain a connection path connecting $\mathbb{L}\mathbb{L}$ day to $\mathbb{L}\mathbb{L}$ last passing through $\mathbb{L}\mathbb{L}$ ordinal-description; 2. we compute the connection path between the words *giorno* and *mese* (i.e., the second branch of the dependency tree) and obtain a path connecting $\mathbb{L}\mathbb{L}$ day to $\mathbb{L}\mathbb{L}$ last passing through $\mathbb{L}\mathbb{L}$ part-of; 3. we compute the overall meaning (\mathcal{M}_O) by composing the connection paths previously computed. In this step the presence of the $\mathbb{L}\mathbb{L}$ ordinal-description concept is detected in the first ontological restriction; moreover $\mathbb{L}\mathbb{L}$ day is recognized as item of this $\mathbb{L}\mathbb{L}$ ordinal-description. At this point we need establishing how $\mathbb{L}\mathbb{L}$ day fits as the smaller part in a $\&$ part-of relation. We scan the remaining ontological restriction(s) by looking for a bigger part involved in a $\&$ part-of relation or in any of its sub-relations. The resulting representation is built by assuming that the larger entity is the reference sequence for the ordering. So, the direct $\mathbb{L}\mathbb{L}$ day-month-part-of of the second ontological restriction is replaced by a path passing through $\mathbb{L}\mathbb{L}$ ordinal-description. In such final ontological restriction (depicted in Figure 8) $\mathbb{L}\mathbb{L}$ day is the $\&$ ord-described-item and $\mathbb{L}\mathbb{L}$ month is the $\&$ reference-sequence.

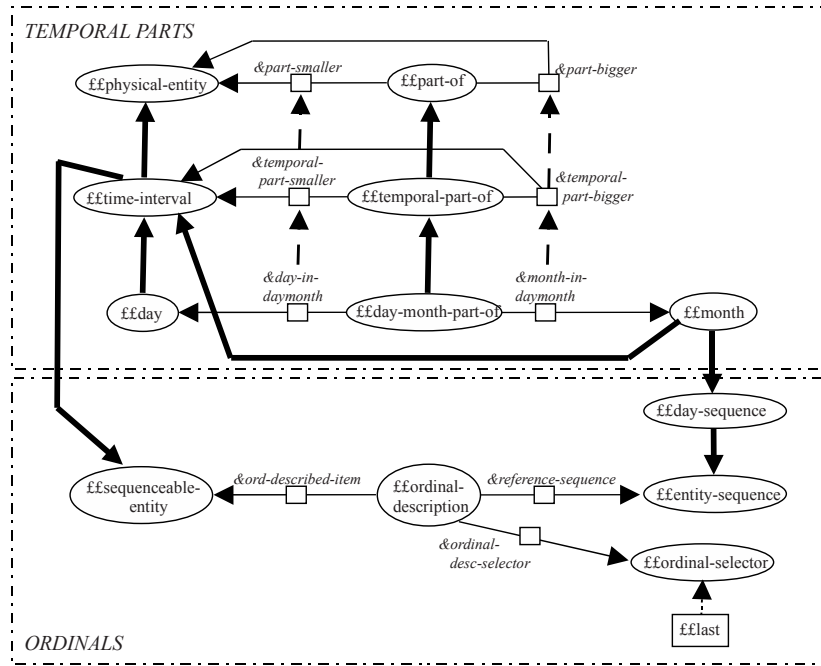


Fig. 7. The fragment of the ontology accounting for ordinals

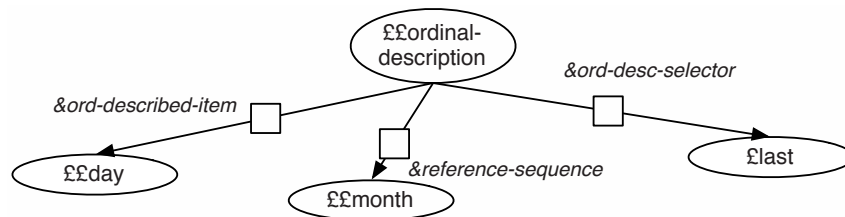


Fig. 8. The resulting ontological restriction produced by the semantic interpreter on the dependency tree in Figure 6

4 Generation

Natural language generation can be described as a three steps process: text planning, sentence planning and realization [10]. Text planning determines which messages to communicate and how rhetorically to structure these messages; sentence planning converts the text plan into a number of sentence plans; realization converts the sentence plans into the final sentences produced. Anyway, in the context of interlingua translation we think that generation needs only for the realization step. Our working hypothesis is that source and target sentences have exactly the same text and the sentence plans. Indeed, the ontological pattern, that is the output of the semantic interpretation step, contains at the same time the information content as well as the rhetorical and sentence structures of the source messages: our choice is to reproduce exactly the same structures in the generation of the target sentences. As a consequence we use a generation system that performs just realization.

In our architecture in the lexicalization we need to account for *lexicalization* too: lexicalization means to choose a particular lexical element (a sign in LIS) in order to express a particular concept. In our architecture we use the OpenCCG realization system [11], an open source tool that has two valuable features: (1) OpenCCG uses *hybrid logic* for semantic structures. Hybrid logic is a kind of propositional modal logic that can be used to represent relational structures [12]: the main feature of hybrid logic is *nominals*, i.e. a new sort of primitive logic elements which explicitly name the nodes of the relational structure. In our project this feature is crucial, since we can represent straightforwardly the ontological pattern produced in the interpretation step in terms of hybrid logic propositions.⁴ (2) OpenCCG applies a *bidirectional grammar* approach, i.e. there is one grammar for both realisation and parsing. It means that derivation and generation have the same structure and that we can develop a grammar by testing its correctness in realization in terms of parsing: as a result, we obtain a speed-up in the process of grammar development [14]. Now we first show how to use hybrid logic to model the ontological path of 8, and second we describe a fragment of a CCG for LIS that is able to generate the target translation.

Ontological restriction and hybrid logic. We can rewrite an ontological restriction by using the interpretation of relational structures (e.g. graphs) in terms of hybrid logic given in [12]: each node of the structure will be represented by a distinct *nominal*, and each edge will be represented by using a distinct *modality label*. Applying this procedure to the ontological restriction of 8 we obtain:

$$\@_{x_0}(\langle\text{ODI}\rangle_{x_1} \wedge \langle\text{ODRS}\rangle_{x_2} \wedge \langle\text{ODS}\rangle_{x_3}) \wedge \@_{x_1}\mathbf{day} \wedge \@_{x_2}\mathbf{month} \wedge \@_{x_3}\mathbf{last} \quad (1)$$

where the nominals x_0, x_1, x_2, x_3 represent the ontological nodes *££ordinal-description*, *££day*, *££month*, *££last* respectively, and the modality labels $\langle\text{ODI}\rangle$, $\langle\text{ODRS}\rangle$, $\langle\text{ODS}\rangle$ represent the ontological relations *&ord-described-item*, *&references-sequence* and *&ordinal-desc-selector* respectively.

A CCG for LIS. The target translation in LIS for the Italian source phrase *ultimo giorno del mese* is *MESE GIORNO ULTIMO*: we now describe how realize this LIS phrase by starting from the hybrid logic formula in equation (1). We developed by hand a CCG for LIS that accounts for a number of morphosyntactic phenomena: in particular we account for morphological realization of plural, spatial agreement between verbs and arguments, coordination [15]. In Tab. 1 we present the fragment of the CCG for the lexical elements involved.

Each element in the grammar has four categories: **LEX**, that contains the lexical form of the item; **PoS**, that contains the part of speech category; **SynCAT**, that contains the syntactic category; **SemCAT**, that contains the semantic category. Note that SynCAT e SemCAT are related by using *semantic variables* (x_i and z_j in Tab. 1): these

⁴ Note that ontological patterns could be written in terms of FOL predicates and, since Hybrid Logic is equivalent to a fragment of FOL, we could rewrite these FOL predicates in terms of hybrid logic, identifying first order variables with nominals of hybrid logic [12]. Moreover our logical interpretation of the ontological pattern does not adhere to the *linguistic meaning* notion that is usually adopted in OpenCCG, i.e. *Hybrid Logic Dependency Semantics* (HLDS) [13].

Table 1. Three lexical elements of the CCG for LIS

LEX	PoS	SynCAT	SemCAT
<i>GIORNO</i>	Noun	n_{x_1}	$@_{x_1} \mathbf{day}$
<i>MESE</i>	Noun	n_{x_0}/n_{z_1}	$@_{x_0} (\langle \text{ODI} \rangle z_1 \wedge \langle \text{ODRS} \rangle x_2) \wedge @_{x_2} \mathbf{month}$
<i>ULTIMO</i>	Adj	$n_{z_2} \setminus n_{z_2}$	$@_{z_2} (\langle \text{ODS} \rangle x_3) @_{x_3} \wedge \mathbf{last}$

<i>MESE</i>	<i>GIORNO</i>	<i>ULTIMO</i>
$n_{x_0}/n_{z_1} : @_{x_0} (\langle \text{ODI} \rangle z_1 \wedge \langle \text{ODRS} \rangle x_2) \wedge @_{x_2} \mathbf{month}$	$n_{x_1} : @_{x_1} \mathbf{day}$	$n_{z_2} \setminus n_{z_2} : @_{z_2} (\langle \text{ODS} \rangle x_3) \wedge @_{x_3} \mathbf{last}$
$>$		
$n_{x_0} : @_{x_0} (\langle \text{ODI} \rangle x_1 \wedge \langle \text{ODRS} \rangle x_2) \wedge @_{x_1} \mathbf{day} \wedge @_{x_2} \mathbf{month}$		
$<$		
$n_{x_0} : @_{x_0} (\langle \text{ODI} \rangle x_1 \wedge \langle \text{ODRS} \rangle x_2 \wedge \langle \text{ODS} \rangle x_3) \wedge @_{x_1} \mathbf{day} \wedge @_{x_2} \mathbf{month} \wedge @_{x_3} \mathbf{last}$		

Fig. 9. The realization/derivation of the LIS phrase *MESE GIORNO ULTIMO* by using the lexicon in Table 1

variables appear in the syntactic categories, but are used as pointers to the semantic categories [13,11]. For instance, in the syntactic category n_{x_0}/n_{z_1} there are two semantic variables: x_0 and z_1 . When syntactic categories combine in a derivation, the semantic variables are unified and the corresponding semantic categories are unified too (see below the derivation reported in Tab. 9). Note that the nominal x_0 is introduced by the lexical item *MESE*: we are assuming that the semantic ordinal structure is introduced by this lexical element. In Fig. 9 we report the realization of the LIS phrase *MESE GIORNO ULTIMO* based on the lexicon in Tab. 1. For sake of clarity, we are going to describe this realization as a derivation: since we are using a bidirectional grammar, realization applies the same rules of derivation but in the reverse order. The derivation consists of two syntactic steps: in the first step the n_{x_0}/n_{z_1} category (corresponding to *MESE*) combines with the n_{x_1} category (corresponding to *GIORNO*) by a forward application producing a new n_{x_0} category; in the second step, the new n_{x_0} category combines with the $n_{z_2} \setminus n_{z_2}$ category (corresponding to *ULTIMO*) by a forward application producing the final n_{x_0} category. In parallel to these two applications, we have that two semantic variables unify. In the first step, the semantic variable z_1 unifies with the semantic variable x_1 , while in the second step the semantic variable z_2 unifies with the semantic variable x_0 . Finally, the last module of our architecture is a virtual actor, i.e. an artificial character, that synthesizes the LIS produced by the OpenCCG.

5 Conclusions and Future Work

In this paper we have presented an architecture for the translation from Italian into Italian Sign Language. The implemented system goes all throughout the translation process: we parse the input sentence, we then extract the meaning representation and generate the LIS. Finally, the synthesis of gestures takes place, and it is performed by a virtual character. The architecture tackles presently a restricted domain, that is it is focussed on weather forecasts. We have briefly described the ontology devised, which encodes the knowledge used by the semantic interpreter and we have illustrated the generation phase, a component of the system that relies on the CCG paradigm and adopts

a hybrid logic approach for the realization proper. We have provided a working example to illustrate both the semantic interpretation phase and the generation phase. Much work still needs to be done at various levels: *i*) the ontology design needs to be refined to fully account for the richness of weather forecasts and related descriptions; *ii*) the semantic interpreter can be improved, e.g. by focussing on the redundancies resulting from the shortest path procedures, and by adding to the ontology some shortcuts to save computational efforts in computing the meaning representation; *iii*) the generation module still needs refinements, as regards as to consider further syntactic phenomena. All described modules will need substantial improvements in order to extend the coverage of the system.

Acknowledgments. This work has been partially supported by the ATLAS project, that is co-funded by Regione Piemonte within the “Converging Technologies - CIPE 2007” framework (Research Sector: Cognitive Science and ICT).

References

1. Jurafsky, D., Martin, J.H.: Speech and Language Processing: An Introduction to Natural Language Processing. In: Computational Linguistics and Speech Recognition. Prentice Hall, Englewood Cliffs (2008)
2. Nirenburg, S., Raskin, V.: Ontological Semantics. MIT Press, Cambridge (2004)
3. Lesmo, L.: The Rule-Based Parser of the NLP Group of the University of Torino. *Intelligenza Artificiale* 2, 46–47 (2007)
4. Miller, G.A., Beckwith, R., Fellbaum, C., Gross, D., Miller, K.J.: Introduction to WordNet: An On-line Lexical Database. *International Journal of Lexicography* 4, 235–244 (1993)
5. Bateman, J.A.: Linguistic interaction and ontological mediation. In: *Ontolinguistics*. In: How Ontological Status Shapes the Linguistic Coding of Concepts. Mouton De Gruyter (2007)
6. Borgo, S., Guarino, N., Masolo, C.: Stratified ontologies: the case of physical objects. In: *Proceedings of the Workshop on Ontology Engineering, ECAI 1996* (1996)
7. Gangemi, A., Mika, P.: Understanding the Semantic Web through Descriptions and Situations. In: Chung, S., Schmidt, D.C. (eds.) *CoopIS 2003, DOA 2003, and ODBASE 2003*. LNCS, vol. 2888, pp. 689–706. Springer, Heidelberg (2003)
8. Gangemi, A.: Norms and plans as unification criteria for social collectives. *Journal of Autonomous Agents and Multi-Agent Systems* 16, 70–112 (2008)
9. Masolo, C., Borgo, S., Gangemi, A., Guarino, N., Oltramari, A., Schneider, L.: *Wonder Web deliverable D17*. Technical Report D17, ISTC-CNR (2002)
10. Reiterand, E., Dale, R.: *Building natural language generation systems*. Cambridge University Press, Cambridge (2000)
11. White, M.: Efficient realization of coordinate structures in combinatory categorial grammar. *Research on Language and Computation* 2006, 39–75 (2006)
12. Blackburn, P.: Representation, reasoning, and relational structures: a hybrid logic manifesto. *Logic Journal of the IGPL* 8, 339–625 (2000)
13. Baldrige, J., Kruijff, G.J.M.: Coupling ccg and hybrid logic dependency semantics. In: *Procs. of ACL 2002*, pp. 319–326. ACL, Morristown (2002)
14. White, M., Clark, R.A.J., Moore, J.D.: Generating tailored, comparative descriptions with contextually appropriate intonation. *Computational Linguistics* 36, 159–201 (2010)
15. Volterra, V. (ed.): *La lingua dei segni italiana*. Il Mulino (2004)